

Unittest for socket_protocol

February 28, 2021

Contents

1	Test Information	4
1.1	Test Candidate Information	4
1.2	Unittest Information	4
1.3	Test System Information	4
2	Statistic	4
2.1	Test-Statistic for testrun with python 2.7.18 (final)	4
2.2	Test-Statistic for testrun with python 3.8.5 (final)	5
2.3	Coverage Statistic	5
3	Tested Requirements	6
3.1	Message Object	6
3.1.1	Status	6
3.1.2	Service-ID	7
3.1.3	Data-ID	8
3.1.4	Data	9
3.2	Communication	10
3.2.1	A full Message Object including the defined properties and data shall be transfered.	10
3.2.2	A checksum shall ensure the correct transmittion	11
3.2.3	An authentication between server and client shall be possible including status feedback methods	12
3.2.4	An automatic authentication shall available	14
3.2.5	Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.	15
3.2.6	A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts	17
3.2.7	Define a channel name for the server and client after connection is established	19
3.2.8	The User shall be able to define a new service	21
3.2.9	Registration of already registered request Service-ID or response Service-ID shall not be possible .	22
3.3	Callbacks	23
3.3.1	It shall be possible to register a callback for a specific Service- and Data-ID	23
3.3.2	It shall be possible to register a callback for a specific Service-ID and all Data-IDs	24

3.3.3	It shall be possible to register a callback for a specific Data-IDs and all Service-IDs	25
3.3.4	It shall be possible to register a callback for all incoming messages	26
3.3.5	Callback choice, if several callbacks are available (caused by wildcard callbacks)	27
3.4	Some additional Information and Passthrough Methods	28
3.4.1	Connection established information	28
3.4.2	Is connected information	29
3.4.3	Reconnect Method	30
3.5	Deprepeated struct protocol	32
3.5.1	A full Message Object including the defined properties and data shall be transfered.	32
A	Trace for testrun with python 2.7.18 (final)	34
A.1	Tests with status Info (22)	34
A.1.1	Status	34
A.1.2	Service-ID	34
A.1.3	Data-ID	35
A.1.4	Data	36
A.1.5	A full Message Object including the defined properties and data shall be transfered.	37
A.1.6	A checksum shall ensure the correct transmission	43
A.1.7	An authentication between server and client shall be possible including status feedback methods	49
A.1.8	An automatic authentication shall available	63
A.1.9	Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.	72
A.1.10	A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts	86
A.1.11	Define a channel name for the server and client after connection is established	96
A.1.12	The User shall be able to define a new service	108
A.1.13	Registration of already registered request Service-ID or response Service-ID shall not be possible .	115
A.1.14	It shall be possible to register a callback for a specific Service- and Data-ID	119
A.1.15	It shall be possible to register a callback for a specific Service-ID and all Data-IDs	129
A.1.16	It shall be possible to register a callback for a specific Data-IDs and all Service-IDs	134
A.1.17	It shall be possible to register a callback for all incoming messages	139
A.1.18	Callback choice, if several callbacks are available (caused by wildcard callbacks)	144

A.1.19	Connection established information	156
A.1.20	Is connected information	166
A.1.21	Reconnect Method	169
A.1.22	A full Message Object including the defined properties and data shall be transferred.	175
B	Trace for testrun with python 3.8.5 (final)	179
B.1	Tests with status Info (22)	179
B.1.1	Status	179
B.1.2	Service-ID	180
B.1.3	Data-ID	181
B.1.4	Data	182
B.1.5	A full Message Object including the defined properties and data shall be transferred.	183
B.1.6	A checksum shall ensure the correct transmission	189
B.1.7	An authentication between server and client shall be possible including status feedback methods	194
B.1.8	An automatic authentication shall available	208
B.1.9	Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.	218
B.1.10	A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts	232
B.1.11	Define a channel name for the server and client after connection is established	242
B.1.12	The User shall be able to define a new service	254
B.1.13	Registration of already registered request Service-ID or response Service-ID shall not be possible .	261
B.1.14	It shall be possible to register a callback for a specific Service- and Data-ID	265
B.1.15	It shall be possible to register a callback for a specific Service-ID and all Data-IDs	274
B.1.16	It shall be possible to register a callback for a specific Data-IDs and all Service-IDs	279
B.1.17	It shall be possible to register a callback for all incoming messages	285
B.1.18	Callback choice, if several callbacks are available (caused by wildcard callbacks)	290
B.1.19	Connection established information	301
B.1.20	Is connected information	311
B.1.21	Reconnect Method	315
B.1.22	A full Message Object including the defined properties and data shall be transferred.	321
C	Test-Coverage	325
C.1	socket_protocol	325
C.1.1	socket_protocol.__init__.py	325

1 Test Information

1.1 Test Candidate Information

The Module `socket_protocol` is designed for point to point communication for client-server issues. For more Information read the sphinx documentation.

Library Information	
Name	socket_protocol
State	Released
Supported Interpreters	python2, python3
Version	20d646cb01f1c24752142245b0c4b36c

Dependencies	
stringtools	09b4d1c41b828c8d1ccb723fa1fd79a9

1.2 Unittest Information

Unittest Information	
Version	da1e6bd881c6d5dd865ce87d511e702f
Testruns with	python 2.7.18 (final), python 3.8.5 (final)

1.3 Test System Information

System Information	
Architecture	64bit
Distribution	Linux Mint 20.1 ulyssa
Hostname	erle
Kernel	5.8.0-44-generic (#50 20.04.1-Ubuntu SMP Wed Feb 10 21:07:30 UTC 2021)
Machine	x86_64
Path	/usr/data/dirk/prj/unittest/socket_protocol/unittest
System	Linux
Username	dirk

2 Statistic

2.1 Test-Statistic for testrun with python 2.7.18 (final)

Number of tests	22
Number of successfull tests	22
Number of possibly failed tests	0
Number of failed tests	0

Executionlevel	Full Test (all defined tests)
Time consumption	19.450s

2.2 Test-Statistic for testrun with python 3.8.5 (final)

Number of tests	22
Number of successfull tests	22
Number of possibly failed tests	0
Number of failed tests	0

Executionlevel	Full Test (all defined tests)
Time consumption	19.341s

2.3 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
socket_protocol	99.5%	100.0%
socket_protocol.__init__.py	99.5%	

3 Tested Requirements

3.1 Message Object

A Message Object shall hold the following information for transmission.

3.1.1 Status

Description

The Status shall hold some general information (in most cases it is used by the responder). Examples: Okay, Service or Data unknown, Operation not permitted, Authentication required, ...

Reason for the implementation

Give the possibility to transfer additional status information (e.g. to explain negative responses).

Fitcriterion

A Status is part of the Message Object and it is holding the Status information.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.1!

Testrun:	python 2.7.18 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (26)
Start-Time:	2021-02-28 18:59:29,954
Finished-Time:	2021-02-28 18:59:29,955
Time-Consumption	0.001s

Testsummary:

Info	Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}
Success	status is part of the message object is correct ('status' is in the list or dict).
Info	Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}
Success	status is part of the message object is correct ('status' is in the list or dict).
Success	Content in message object for status is correct (Content 'S' and Type is <type 'str'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.1!

Testrun:	python 3.8.5 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (26)
Start-Time:	2021-02-28 18:59:51,634
Finished-Time:	2021-02-28 18:59:51,636
Time-Consumption	0.001s

Testsummary:

Info	Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}
Success	status is part of the message object is correct ('status' is in the list or dict).
Info	Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}
Success	status is part of the message object is correct ('status' is in the list or dict).
Success	Content in message object for status is correct (Content 'S' and Type is <class 'str'>).

3.1.2 Service-ID

Description

The Service-ID shall hold information about the type of the request / corresponding response. Examples: read request, write request, read response, write response, ...

Reason for the implementation

Give the requestor the possibility to use different types (Services) for a transfer.

Fitcriterion

A Service-ID is part of the Message Object and it is holding the Service-ID information.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.2!

Testrun:	python 2.7.18 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (27)
Start-Time:	2021-02-28 18:59:29,955
Finished-Time:	2021-02-28 18:59:29,956
Time-Consumption	0.001s

Testsummary:

Info	Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}
Success	service_id is part of the message object is correct ('service_id' is in the list or dict).
Info	Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}
Success	service_id is part of the message object is correct ('service_id' is in the list or dict).
Success	Content in message object for service_id is correct (Content 'SID' and Type is <type 'str'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.2!

Testrun:	python 3.8.5 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (27)
Start-Time:	2021-02-28 18:59:51,636
Finished-Time:	2021-02-28 18:59:51,636

Time-Consumption 0.001s

Testsummary:

Info Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}
Success service_id is part of the message object is correct ('service_id' is in the list or dict).
Info Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}
Success service_id is part of the message object is correct ('service_id' is in the list or dict).
Success Content in message object for service_id is correct (Content 'SID' and Type is <class 'str'>).

3.1.3 Data-ID

Description

The Data-ID shall hold information to differtiate the data for a specific Service.

Reason for the implementation

Give the possibility to transfer different information for each Service.

Fitcriterion

A Data-ID is part of the Message Object and it is holding the Data-ID information.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.3!

Testrun: python 2.7.18 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init....py (28)
 Start-Time: 2021-02-28 18:59:29,956
 Finished-Time: 2021-02-28 18:59:29,957
 Time-Consumption 0.001s

Testsummary:

Info Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}
Success data_id is part of the message object is correct ('data_id' is in the list or dict).
Info Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}
Success data_id is part of the message object is correct ('data_id' is in the list or dict).
Success Content in message object for data_id is correct (Content 'DID' and Type is <type 'str'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.3!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init....py (28)
 Start-Time: 2021-02-28 18:59:51,637

Finished-Time: 2021-02-28 18:59:51,637
 Time-Consumption 0.001s

Testsummary:

Info Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}
Success data_id is part of the message object is correct ('data_id' is in the list or dict).
Info Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}
Success data_id is part of the message object is correct ('data_id' is in the list or dict).
Success Content in message object for data_id is correct (Content 'DID' and Type is <class 'str'>).

3.1.4 Data

Description

The Data shall hold the data to be transfered. For the most requests not data is transmitted.

Reason for the implementation

Give the possibility to transfer Data.

Fitcriterion

Data is part of the Message Object and it is holding the Data information.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.4!

Testrun: python 2.7.18 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (29)
 Start-Time: 2021-02-28 18:59:29,957
 Finished-Time: 2021-02-28 18:59:29,957
 Time-Consumption 0.001s

Testsummary:

Info Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}
Success data is part of the message object is correct ('data' is in the list or dict).
Info Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}
Success data is part of the message object is correct ('data' is in the list or dict).
Success Content in message object for data is correct (Content 'D' and Type is <type 'str'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.4!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (29)

Start-Time: 2021-02-28 18:59:51,637
 Finished-Time: 2021-02-28 18:59:51,638
 Time-Consumption 0.001s

Testsummary:

Info Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}
Success data is part of the message object is correct ('data' is in the list or dict).
Info Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}
Success data is part of the message object is correct ('data' is in the list or dict).
Success Content in message object for data is correct (Content 'D' and Type is <class 'str'>).

3.2 Communication

3.2.1 A full Message Object including the defined properties and data shall be transfered.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.5!

Testrun: python 2.7.18 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_...py (33)
 Start-Time: 2021-02-28 18:59:29,958
 Finished-Time: 2021-02-28 18:59:30,715
 Time-Consumption 0.757s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info Transferring a message client → server
Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).
Success Received message on server side is correct (Content {'u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transfered', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).
Info Transferring a message server → client
Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).
Success Received message on client side is correct (Content {'u'status': 4, u'service_id': 17, u'data': u'msg2_data_to_be_transfered', u'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.5!

Testrun:	python 3.8.5 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (33)
Start-Time:	2021-02-28 18:59:51,638
Finished-Time:	2021-02-28 18:59:52,394
Time-Consumption	0.756s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).

3.2.2 A checksum shall ensure the correct transmittion**Description**

If the checksum does not fit to the checksum of the transferred data, the message will be ignored, because the complete content including the Service- and Data-ID is possibly corrupted.

Reason for the implementation

Ensure correct data transfer.

Fitcriterion

Corrupted message is not in the receive buffer after transmission.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.6!

Testrun:	python 2.7.18 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (34)
Start-Time:	2021-02-28 18:59:30,716
Finished-Time:	2021-02-28 18:59:31,690
Time-Consumption	0.974s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success Checksum Error → No message received by server is correct (Content None and Type is <type 'NoneType'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Success Checksum Error → No message received by client is correct (Content None and Type is <type 'NoneType'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.6!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (34)
 Start-Time: 2021-02-28 18:59:52,394
 Finished-Time: 2021-02-28 18:59:53,357
 Time-Consumption 0.963s

Testsummary:

Info Setting up communication

Info Connecting Server and Client

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

Success Checksum Error → No message received by server is correct (Content None and Type is <class 'NoneType'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Success Checksum Error → No message received by client is correct (Content None and Type is <class 'NoneType'>).

3.2.3 An authentication between server and client shall be possible including status feedback methods

Description

The Client shall have a method to initiate the authentication. In case that the server and the client do have identical secrets, the authentication shall be successfull.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

Check authentication method feedback (client) and authentication feedback (client and server), in case of differing and identical secrets.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.7!

Testrun: python 2.7.18 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (35)

Start-Time: 2021-02-28 18:59:31,691
 Finished-Time: 2021-02-28 18:59:32,772
 Time-Consumption 1.081s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info No secret set
Info Performing Authentication
Success Return Value of authentication method is correct (Content False and Type is <type 'bool'>).
Success Authentication state of server is correct (Content True and Type is <type 'bool'>).
Success Authentication state of client is correct (Content True and Type is <type 'bool'>).
Info Different secrets set
Success Authentication state of server is correct (Content False and Type is <type 'bool'>).
Success Authentication state of client is correct (Content False and Type is <type 'bool'>).
Info Performing Authentication
Success Return Value of authentication method is correct (Content False and Type is <type 'bool'>).
Success Authentication state of server is correct (Content False and Type is <type 'bool'>).
Success Authentication state of client is correct (Content False and Type is <type 'bool'>).
Info Identical secrets set
Info Performing Authentication
Success Return Value of authentication method is correct (Content True and Type is <type 'bool'>).
Success Authentication state of server is correct (Content True and Type is <type 'bool'>).
Success Authentication state of client is correct (Content True and Type is <type 'bool'>).
Info Corrupting the authentication mechanism
Info Performing Authentication
Success Return Value of authentication method is correct (Content False and Type is <type 'bool'>).
Success Authentication state of server is correct (Content False and Type is <type 'bool'>).
Success Authentication state of client is correct (Content False and Type is <type 'bool'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.7!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init....py (35)
 Start-Time: 2021-02-28 18:59:53,358
 Finished-Time: 2021-02-28 18:59:54,329
 Time-Consumption 0.972s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info No secret set
Info Performing Authentication
Success Return Value of authentication method is correct (Content False and Type is <class 'bool'>).
Success Authentication state of server is correct (Content True and Type is <class 'bool'>).
Success Authentication state of client is correct (Content True and Type is <class 'bool'>).
Info Different secrets set

```

Success Authentication state of server is correct (Content False and Type is <class 'bool'>).
Success Authentication state of client is correct (Content False and Type is <class 'bool'>).
Info Performing Authentication
Success Return Value of authentication method is correct (Content False and Type is <class 'bool'>).
Success Authentication state of server is correct (Content False and Type is <class 'bool'>).
Success Authentication state of client is correct (Content False and Type is <class 'bool'>).
Info Identical secrets set
Info Performing Authentication
Success Return Value of authentication method is correct (Content True and Type is <class 'bool'>).
Success Authentication state of server is correct (Content True and Type is <class 'bool'>).
Success Authentication state of client is correct (Content True and Type is <class 'bool'>).
Info Corrupting the authentication mechanism
Info Performing Authentication
Success Return Value of authentication method is correct (Content False and Type is <class 'bool'>).
Success Authentication state of server is correct (Content False and Type is <class 'bool'>).
Success Authentication state of client is correct (Content False and Type is <class 'bool'>).
    
```

3.2.4 An automatic authentication shall available

Description

An authentication is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentication.

Fitcriterion

Check authentication feedback (client and server) after connect has been triggered.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.8!

```

Testrun:      python 2.7.18 (final)
Caller:      /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (36)
Start-Time:  2021-02-28 18:59:32,772
Finished-Time: 2021-02-28 18:59:35,493
Time-Consumption 2.720s
    
```

Testsummary:

```

Info Setting up communication
Info Connecting Server and Client
Info Identical secrets set and automatic authentication
Success Authentication state of server is correct (Content False and Type is <type 'bool'>).
Success Authentication state of client is correct (Content False and Type is <type 'bool'>).
Info Connecting Server and Client
Success Authentication state of server is correct (Content True and Type is <type 'bool'>).
Success Authentication state of client is correct (Content True and Type is <type 'bool'>).
    
```

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.8!

Testrun:	python 3.8.5 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (36)
Start-Time:	2021-02-28 18:59:54,330
Finished-Time:	2021-02-28 18:59:57,044
Time-Consumption	2.714s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Identical secrets set and automatic authentication
Success	Authentication state of server is correct (Content False and Type is <class 'bool'>).
Success	Authentication state of client is correct (Content False and Type is <class 'bool'>).
Info	Connecting Server and Client
Success	Authentication state of server is correct (Content True and Type is <class 'bool'>).
Success	Authentication state of client is correct (Content True and Type is <class 'bool'>).

3.2.5 Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.

Description

Communication (rx and tx) shall be disabled, if a secret is given. Except of a response for registered services, saying that a Authentication is required.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

RX and TX is not possible, till a successfull authentication has been performed.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.9!

Testrun:	python 2.7.18 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (37)
Start-Time:	2021-02-28 18:59:35,494
Finished-Time:	2021-02-28 18:59:37,485
Time-Consumption	1.991s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Setting a Server secret and no Client secret
Info	Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on server side is correct (Content {u'status': 3, u'service_id': 31, u'data': None, u'data_id': 36} and Type is <class 'socket_protocol.data_storage'>).

Info Setting no Server secret but a Client secret

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Info Identical secrets set

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content False and Type is <type 'bool'>).

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Info Performing Authentication

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on client side is correct (Content {u'status': 4, u'service_id': 17, u'data': u'msg2_data_to_be_transferred', u'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.9!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (37)
 Start-Time: 2021-02-28 18:59:57,045
 Finished-Time: 2021-02-28 18:59:59,027
 Time-Consumption 1.982s

Testsummary:

Info Setting up communication

Info Connecting Server and Client

Info Setting a Server secret and no Client secret

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

Success Received message on server side is correct (Content {'data_id': 36, 'service_id': 31, 'status': 3, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

Info Setting no Server secret but a Client secret

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

Info Identical secrets set

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content False and Type is <class 'bool'>).
Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).
Info Transferring a message server → client
Success Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).
Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).
Info Performing Authentication
Info Transferring a message client → server
Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).
Info Transferring a message server → client
Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).
Success Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).

3.2.6 A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts

Description

It shall be possible to add a specific message, identified by Service-ID and Data-ID, to a whitelist. All messages added to that whitelist shall be transmitted and received, if no authentication was successful performed.

Reason for the implementation

Give the user the possibility to define messages which will not be protected behind the authentication mechanism.

Fitcriterion

Transmission and Reception will be enabled, after the message has been added to the whitelist.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.10!

Testrun:	python 2.7.18 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (38)
Start-Time:	2021-02-28 18:59:37,486
Finished-Time:	2021-02-28 18:59:39,981
Time-Consumption	2.495s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info Identical secrets set
Info Transferring a message client → server
Success Returnvalue of Client send Method is correct (Content False and Type is <type 'bool'>).
Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).
Info Transferring a message server → client
Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Info Added msg1 to client whitelist (sid=17, did=34)

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Info Added msg1 to server whitelist (sid=17, did=34)

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Info Added msg2 to client and server whitelist (sid=17, did=35)

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on client side is correct (Content {u'status': 4, u'service_id': 17, u'data': u'msg2_data_to_be_transferred', u'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.10!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (38)
 Start-Time: 2021-02-28 18:59:59,028
 Finished-Time: 2021-02-28 19:00:01,520
 Time-Consumption 2.492s

Testsummary:

Info Setting up communication

Info Connecting Server and Client

Info Identical secrets set

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content False and Type is <class 'bool'>).

Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

```

Info           Added msg1 to client whitelist (sid=17, did=34)
Info           Transferring a message client → server
Success       Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success       Received message on server side is correct (Content None and Type is <class 'NoneType'>).
Info           Transferring a message server → client
Success       Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).
Success       Received message on client side is correct (Content None and Type is <class 'NoneType'>).
Info           Added msg1 to server whitelist (sid=17, did=34)
Info           Transferring a message client → server
Success       Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success       Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0,
'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).
Info           Transferring a message server → client
Success       Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).
Success       Received message on client side is correct (Content None and Type is <class 'NoneType'>).
Info           Added msg2 to client and server whitelist (sid=17, did=35)
Info           Transferring a message client → server
Success       Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success       Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0,
'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).
Info           Transferring a message server → client
Success       Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).
Success       Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4,
'data': 'msg2_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).
    
```

3.2.7 Define a channel name for the server and client after connection is established

Description

After the connection is established, the client will initiate the channel name exchange. The channel name defined on the client side will be dominant.

Reason for the implementation

Structured logging by creating logger childs for each channel.

Fitcriterion

Perform a channel name exchange with no channel name definition, differing channel name definition and identical channel name definition. In all cases, the channel name of the client will be used. Perform two channel name exchanges with only one channel name definition. This definition will be used.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.11!

```

Testrun:         python 2.7.18 (final)
Caller:          /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (39)
Start-Time:     2021-02-28 18:59:39,982
Finished-Time:  2021-02-28 18:59:41,746
    
```

Time-Consumption 1.764s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info Setting no Channel name for server and client
Success Channel name of server is correct (Content None and Type is <type 'NoneType'>).
Success Channel name of client is correct (Content None and Type is <type 'NoneType'>).
Info Setting different Channel names for client and Server
Info Connecting Server and Client
Success Channel name of server is correct (Content 'client' and Type is <type 'str'>).
Success Channel name of client is correct (Content 'client' and Type is <type 'str'>).
Info Setting identical Channel names for client and server
Info Connecting Server and Client
Success Channel name of server is correct (Content 'unittest' and Type is <type 'str'>).
Success Channel name of client is correct (Content 'unittest' and Type is <type 'str'>).
Info Setting Channel name for client only
Info Connecting Server and Client
Success Channel name of server is correct (Content 'client' and Type is <type 'str'>).
Success Channel name of client is correct (Content 'client' and Type is <type 'str'>).
Info Setting Channel name for server only
Info Connecting Server and Client
Success Channel name of server is correct (Content 'server' and Type is <type 'str'>).
Success Channel name of client is correct (Content 'server' and Type is <type 'str'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.11!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (39)
 Start-Time: 2021-02-28 19:00:01,521
 Finished-Time: 2021-02-28 19:00:03,274
 Time-Consumption 1.753s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info Setting no Channel name for server and client
Success Channel name of server is correct (Content None and Type is <class 'NoneType'>).
Success Channel name of client is correct (Content None and Type is <class 'NoneType'>).
Info Setting different Channel names for client and Server
Info Connecting Server and Client
Success Channel name of server is correct (Content 'client' and Type is <class 'str'>).
Success Channel name of client is correct (Content 'client' and Type is <class 'str'>).
Info Setting identical Channel names for client and server
Info Connecting Server and Client
Success Channel name of server is correct (Content 'unittest' and Type is <class 'str'>).
Success Channel name of client is correct (Content 'unittest' and Type is <class 'str'>).

Info Setting Channel name for client only
Info Connecting Server and Client
Success Channel name of server is correct (Content 'client' and Type is <class 'str'>).
Success Channel name of client is correct (Content 'client' and Type is <class 'str'>).
Info Setting Channel name for server only
Info Connecting Server and Client
Success Channel name of server is correct (Content 'server' and Type is <class 'str'>).
Success Channel name of client is correct (Content 'server' and Type is <class 'str'>).

3.2.8 The User shall be able to define a new service

Description

The service is defined by a Request Service-ID and a Response Service-ID.

Reason for the implementation

Definition of Request and Response SIDs.

Fitcriterion

Define a service and check, that the server will respond on the new Service-ID. The Status shall be "Request has no callback. Data buffered.", because no callback is registered for that request.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.12!

Testrun:	python 2.7.18 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (40)
Start-Time:	2021-02-28 18:59:41,746
Finished-Time:	2021-02-28 18:59:42,612
Time-Consumption	0.866s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info Transferring a message client → server → client
Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).
Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).
Info Adding service to server instance for the transmit message
Info Transferring a message client → server → client
Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).
Success Received message on server side is correct (Content {u'status': 1, u'service_id': 18, u'data': None, u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.12!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (40)
 Start-Time: 2021-02-28 19:00:03,275
 Finished-Time: 2021-02-28 19:00:04,141
 Time-Consumption 0.866s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info Transferring a message client → server → client
Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).
Info Adding service to server instance for the transmit message
Info Transferring a message client → server → client
Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 18, 'status': 1, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

3.2.9 Registration of already registered request Service-ID or response Service-ID shall not be possible

Description

An exception shall be raised, if a service registration with an existing request SID or response SID is performed.

Reason for the implementation

Changing existing services will create strange situations with already registered callbacks.

Fitcriterion

Catch exception for registration of existing request and response SID.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.13!

Testrun: python 2.7.18 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (41)
 Start-Time: 2021-02-28 18:59:42,614
 Finished-Time: 2021-02-28 18:59:42,969
 Time-Consumption 0.355s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info Adding a service with an already registered request SID
Success Expected Exception RequestSidExistsError was triggered
Info Adding a service with an already registered response SID
Success Expected Exception ResponseSidExistsError was triggered

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.13!

Testrun:	python 3.8.5 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (41)
Start-Time:	2021-02-28 19:00:04,141
Finished-Time:	2021-02-28 19:00:04,502
Time-Consumption	0.361s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Adding a service with an already registered request SID
Success	Expected Exception RequestSidExistsError was triggered
Info	Adding a service with an already registered response SID
Success	Expected Exception ResponseSidExistsError was triggered

3.3 Callbacks**3.3.1 It shall be possible to register a callback for a specific Service- and Data-ID****Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.14!

Testrun:	python 2.7.18 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (45)
Start-Time:	2021-02-28 18:59:42,970
Finished-Time:	2021-02-28 18:59:43,942
Time-Consumption	0.973s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Info	Overwriting existing Callback using one with faulty (too many) return values
Info	Transferring data
Success	Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {u'status': 2, u'service_id': 11, u'data': None, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing the registered Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content None and Type is <type 'NoneType'>).

Success Message received by client is correct (Content {u'status': 1, u'service_id': 11, u'data': None, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.14!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (45)
 Start-Time: 2021-02-28 19:00:04,503
 Finished-Time: 2021-02-28 19:00:05,472
 Time-Consumption 0.969s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info Registering a correct working Callback
Info Transferring data
Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).
Info Overwriting existing Callback using one with faulty (too many) return values
Info Transferring data
Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 2, 'data': None} and Type is <class 'socket_protocol.data_storage'>).
Info Removing the registered Callback
Info Transferring data
Success Message stored inside callback is correct (Content None and Type is <class 'NoneType'>).
Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 1, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

3.3.2 It shall be possible to register a callback for a specific Service-ID and all Data-IDs

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.15!

Testrun: python 2.7.18 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (46)
 Start-Time: 2021-02-28 18:59:43,943
 Finished-Time: 2021-02-28 18:59:44,500
 Time-Consumption 0.556s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info Registering a correct working Callback

Info Transferring data
Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.15!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (46)
 Start-Time: 2021-02-28 19:00:05,472
 Finished-Time: 2021-02-28 19:00:06,035
 Time-Consumption 0.562s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info Registering a correct working Callback
Info Transferring data
Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

3.3.3 It shall be possible to register a callback for a specific Data-IDs and all Service-IDs

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.16!

Testrun: python 2.7.18 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (47)
 Start-Time: 2021-02-28 18:59:44,500
 Finished-Time: 2021-02-28 18:59:45,058
 Time-Consumption 0.558s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Info Registering a correct working Callback
Info Transferring data
Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.16!

Testrun:	python 3.8.5 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (47)
Start-Time:	2021-02-28 19:00:06,035
Finished-Time:	2021-02-28 19:00:06,596
Time-Consumption	0.560s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data.id': 0, 'service.id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data.id': 0, 'service.id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

3.3.4 It shall be possible to register a callback for all incoming messages

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.17!

Testrun:	python 2.7.18 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (48)
Start-Time:	2021-02-28 18:59:45,059
Finished-Time:	2021-02-28 18:59:45,618
Time-Consumption	0.559s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {'status': 0, 'service.id': 10, 'data': 31, 'data.id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'status': 0, 'service.id': 11, 'data': 33, 'data.id': 0} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.17!

Testrun:	python 3.8.5 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (48)
Start-Time:	2021-02-28 19:00:06,596
Finished-Time:	2021-02-28 19:00:07,157
Time-Consumption	0.560s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

3.3.5 Callback choice, if several callbacks are available (caused by wildcard callbacks)**Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.18!

Testrun:	python 2.7.18 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (49)
Start-Time:	2021-02-28 18:59:45,619
Finished-Time:	2021-02-28 18:59:46,791
Time-Consumption	1.172s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering all kind of Callbacks
Info	Transferring data
Success	Message stored inside callback is correct (Content {'status': 0, 'service_id': 10, 'data': 31, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'status': 0, 'service_id': 11, 'data': 33, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Data- and Service-ID
Info	Transferring data
Success	Message stored inside callback is correct (Content {'status': 0, 'service_id': 10, 'data': 31, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'status': 6, 'service_id': 11, 'data': 34, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Service-ID and all Data-IDs
Info	Transferring data
Success	Message stored inside callback is correct (Content {'status': 0, 'service_id': 10, 'data': 31, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'status': 6, 'service_id': 11, 'data': 35, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Data-ID and all Service-IDs
Info	Transferring data
Success	Message stored inside callback is correct (Content {'status': 0, 'service_id': 10, 'data': 31, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'status': 0, 'service_id': 11, 'data': 36, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.18!

Testrun:	python 3.8.5 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (49)
Start-Time:	2021-02-28 19:00:07,158
Finished-Time:	2021-02-28 19:00:08,334
Time-Consumption	1.177s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering all kind of Callbacks
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data.id': 0, 'service.id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data.id': 0, 'service.id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Data- and Service-ID
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data.id': 0, 'service.id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data.id': 0, 'service.id': 11, 'status': 6, 'data': 34} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Service-ID and all Data-IDs
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data.id': 0, 'service.id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data.id': 0, 'service.id': 11, 'status': 6, 'data': 35} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Data-ID and all Service-IDs
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data.id': 0, 'service.id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data.id': 0, 'service.id': 11, 'status': 0, 'data': 36} and Type is <class 'socket_protocol.data_storage'>).

3.4 Some additional Information and Passthrough Methods**3.4.1 Connection established information****Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.19!

Testrun:	python 2.7.18 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (53)
Start-Time:	2021-02-28 18:59:46,792
Finished-Time:	2021-02-28 18:59:47,603
Time-Consumption	0.811s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Success	Client connection status is correct (Content True and Type is <type 'bool'>).
Success	Server connection status is correct (Content True and Type is <type 'bool'>).
Success	Client connection status is correct (Content False and Type is <type 'bool'>).
Success	Server connection status is correct (Content False and Type is <type 'bool'>).
Info	Connecting Server and Client
Success	Client connection status is correct (Content True and Type is <type 'bool'>).
Success	Server connection status is correct (Content True and Type is <type 'bool'>).
Info	Adding secrets to socket_protocol
Success	Client connection status is correct (Content False and Type is <type 'bool'>).
Success	Server connection status is correct (Content False and Type is <type 'bool'>).
Info	Doing authentication
Success	Client connection status is correct (Content True and Type is <type 'bool'>).
Success	Server connection status is correct (Content True and Type is <type 'bool'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.19!

Testrun:	python 3.8.5 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (53)
Start-Time:	2021-02-28 19:00:08,335
Finished-Time:	2021-02-28 19:00:09,148
Time-Consumption	0.812s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Success	Client connection status is correct (Content True and Type is <class 'bool'>).
Success	Server connection status is correct (Content True and Type is <class 'bool'>).
Success	Client connection status is correct (Content False and Type is <class 'bool'>).
Success	Server connection status is correct (Content False and Type is <class 'bool'>).
Info	Connecting Server and Client
Success	Client connection status is correct (Content True and Type is <class 'bool'>).
Success	Server connection status is correct (Content True and Type is <class 'bool'>).
Info	Adding secrets to socket_protocol
Success	Client connection status is correct (Content False and Type is <class 'bool'>).
Success	Server connection status is correct (Content False and Type is <class 'bool'>).
Info	Doing authentication
Success	Client connection status is correct (Content True and Type is <class 'bool'>).
Success	Server connection status is correct (Content True and Type is <class 'bool'>).

3.4.2 Is connected information

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.20!

Testrun: python 2.7.18 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (54)
 Start-Time: 2021-02-28 18:59:47,603
 Finished-Time: 2021-02-28 18:59:47,958
 Time-Consumption 0.355s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Success Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).
Success Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).
Info Disconnecting Server and Client
Success Client Communication instance connection status is correct (Content False and Type is <type 'bool'>).
Success Server Communication instance connection status is correct (Content False and Type is <type 'bool'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.20!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (54)
 Start-Time: 2021-02-28 19:00:09,148
 Finished-Time: 2021-02-28 19:00:09,508
 Time-Consumption 0.359s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Success Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).
Success Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).
Info Disconnecting Server and Client
Success Client Communication instance connection status is correct (Content False and Type is <class 'bool'>).
Success Server Communication instance connection status is correct (Content False and Type is <class 'bool'>).

3.4.3 Reconnect Method

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.21!

Testrun: python 2.7.18 (final)

Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (55)
 Start-Time: 2021-02-28 18:59:47,958
 Finished-Time: 2021-02-28 18:59:48,662
 Time-Consumption 0.703s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Success Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).
Success Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).
Info Disconnecting Server and Client
Success Client Communication instance connection status is correct (Content False and Type is <type 'bool'>).
Success Server Communication instance connection status is correct (Content False and Type is <type 'bool'>).
Info Connecting Server and Client
Success Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).
Success Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.21!

Testrun: python 3.8.5 (final)
 Caller: /usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (55)
 Start-Time: 2021-02-28 19:00:09,508
 Finished-Time: 2021-02-28 19:00:10,218
 Time-Consumption 0.710s

Testsummary:

Info Setting up communication
Info Connecting Server and Client
Success Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).
Success Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).
Info Disconnecting Server and Client
Success Client Communication instance connection status is correct (Content False and Type is <class 'bool'>).
Success Server Communication instance connection status is correct (Content False and Type is <class 'bool'>).
Info Connecting Server and Client
Success Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).
Success Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).

3.5 Deprecaeted struct protocol

3.5.1 A full Message Object including the defined properties and data shall be transfered.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.22!

Testrun:	python 2.7.18 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init....py (59)
Start-Time:	2021-02-28 18:59:48,663
Finished-Time:	2021-02-28 18:59:49,419
Time-Consumption	0.757s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).
Success	Received message on server side is correct (Content {'status': 0, 'service_id': 17, 'data': u'msg1_data_to_be_transfered', 'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).
Success	Received message on client side is correct (Content {'status': 4, 'service_id': 17, 'data': u'msg2_data_to_be_transfered', 'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.22!

Testrun:	python 3.8.5 (final)
Caller:	/usr/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init....py (59)
Start-Time:	2021-02-28 19:00:10,219
Finished-Time:	2021-02-28 19:00:10,988
Time-Consumption	0.770s

Testsummary:

Info	Setting up communication
-------------	--------------------------

Info Connecting Server and Client

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Success Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

A Trace for testrun with python 2.7.18 (final)

A.1 Tests with status Info (22)

A.1.1 Status

Description

The Status shall hold some general information (in most cases it is used by the responder). Examples: Okay, Service or Data unknown, Operation not permitted, Authentication required, ...

Reason for the implementation

Give the possibility to transfer additional status information (e.g. to explain negative responses).

Fitcriterion

A Status is part of the Message Object and it is holding the Status information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}

Success status is part of the message object is correct ('status' is in the list or dict).

Result (status is part of the message object): {'status': None, 'service_id': None, 'data': None, 'data_id': None} (<class 'socket_protocol.data_storage'>)

Expectation (status is part of the message object): 'status' in result

Info Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}

Success status is part of the message object is correct ('status' is in the list or dict).

Result (status is part of the message object): {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'} (<class 'socket_protocol.data_storage'>)

Expectation (status is part of the message object): 'status' in result

Success Content in message object for status is correct (Content 'S' and Type is <type 'str'>).

Result (Content in message object for status): 'S' (<type 'str'>)

Expectation (Content in message object for status): result = 'S' (<type 'str'>)

A.1.2 Service-ID

Description

The Service-ID shall hold information about the type of the request / corresponding response. Examples: read request, write request, read response, write response, ...

Reason for the implementation

Give the requestor the possibility to use different types (Services) for a transfer.

Fitcriterion

A Service-ID is part of the Message Object and it is holding the Service-ID information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}

Success service_id is part of the message object is correct ('service_id' is in the list or dict).

Result (service_id is part of the message object): {'status': None, 'service_id': None, 'data': None, 'data_id': None} (<class 'socket_protocol.data_storage'>)

Expectation (service_id is part of the message object): 'service_id' in result

Info Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}

Success service_id is part of the message object is correct ('service_id' is in the list or dict).

Result (service_id is part of the message object): {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'} (<class 'socket_protocol.data_storage'>)

Expectation (service_id is part of the message object): 'service_id' in result

Success Content in message object for service_id is correct (Content 'SID' and Type is <type 'str'>).

Result (Content in message object for service_id): 'SID' (<type 'str'>)

Expectation (Content in message object for service_id): result = 'SID' (<type 'str'>)

A.1.3 Data-ID

Description

The Data-ID shall hold information to differtiate the data for a specific Service.

Reason for the implementation

Give the possibility to transfer different information for each Service.

Fitcriterion

A Data-ID is part of the Message Object and it is holding the Data-ID information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}

Success data_id is part of the message object is correct ('data_id' is in the list or dict).

Result (data_id is part of the message object): {'status': None, 'service_id': None, 'data':
 ↪ None, 'data_id': None} (<class 'socket_protocol.data_storage'>)

Expectation (data_id is part of the message object): 'data_id' in result

Info Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}

Success data_id is part of the message object is correct ('data_id' is in the list or dict).

Result (data_id is part of the message object): {'status': 'S', 'service_id': 'SID', 'data':
 ↪ 'D', 'data_id': 'DID'} (<class 'socket_protocol.data_storage'>)

Expectation (data_id is part of the message object): 'data_id' in result

Success Content in message object for data_id is correct (Content 'DID' and Type is <type 'str'>).

Result (Content in message object for data_id): 'DID' (<type 'str'>)

Expectation (Content in message object for data_id): result = 'DID' (<type 'str'>)

A.1.4 Data

Description

The Data shall hold the data to be transferred. For the most requests not data is transmitted.

Reason for the implementation

Give the possibility to transfer Data.

Fitcriterion

Data is part of the Message Object and it is holding the Data information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}

Success data is part of the message object is correct ('data' is in the list or dict).

Result (data is part of the message object): {'status': None, 'service_id': None, 'data':
 ↪ None, 'data_id': None} (<class 'socket_protocol.data_storage'>)

Expectation (data is part of the message object): 'data' in result

Info Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}

Success data is part of the message object is correct ('data' is in the list or dict).

Result (data is part of the message object): {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'} (<class 'socket_protocol.data_storage'>)

Expectation (data is part of the message object): 'data' in result

Success Content in message object for data is correct (Content 'D' and Type is <type 'str'>).

Result (Content in message object for data): 'D' (<type 'str'>)

Expectation (Content in message object for data): result = 'D' (<type 'str'>)

A.1.5 A full Message Object including the defined properties and data shall be transfered.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incomming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
```

```
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```



```

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred"
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on server side): {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on server side): result = {'status': 0, 'service_id': 17, 'data': 'msg1_data_to_be_transferred', 'data_id': 34} (<class 'socket_protocol.data_storage'>)

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 35 7d 20 18 19 e8 3a 3e
```

```
comm-client: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 35 7d 20 18 19 e8 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 35 7d 20 18 19 e8
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "u'msg2_data_to_be_transferred'"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): True (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)

Success Received message on client side is correct (Content {'status': 4, 'service_id': 17, 'data': u'msg2_data_to_be_transferred', 'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {'status': 4, 'service_id': 17, 'data':
 ↪ u'msg2_data_to_be_transferred', 'data_id': 35} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on client side): result = {'status': 4, 'service_id': 17,
 ↪ 'data': 'msg2_data_to_be_transferred', 'data_id': 35} (<class
 ↪ 'socket_protocol.data_storage'>)

A.1.6 A checksum shall ensure the correct transmittion

Description

If the checksum does not fit to the checksum of the transferred data, the message will be ignored, because the complete content including the Service- and Data-ID is possibly corrupted.

Reason for the implementation

Ensure correct data transfer.

Fitcriterion

Corrupted message is not in the receive buffer after transmission.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incomming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and

↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the

↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the

↪ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the

↪ authentication whitelist

Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
```

```
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: Cleaning up receive-buffer
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
prot-server: Cleaning up receive-buffer
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9c 3a 3e
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9c
prot-server: Received message has an invalid checksum. Message will be ignored.
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.

```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Checksum Error → No message received by server is correct (Content None and Type is <type 'NoneType'>).

Result (Checksum Error -> No message received by server): None (<type 'NoneType'>)

Expectation (Checksum Error -> No message received by server): result = None (<type 'NoneType'>)

Info Transferring a message server → client

```

prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"

```


Unittest for socket_protocol

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61  
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61  
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20  
↳ 33 35 7d 20 18 19 e8 3a 3e
```

```
comm-client: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20  
↳ 33 35 7d 20 18 19 e8 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61  
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a  
↳ 20 33 35 7d 20 18 19 e8
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:  
↳ "u'msg2_data_to_be_transferred"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive  
↳ method
```

```
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not  
↳ in buffer.
```

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

```
Result (Returnvalue of Server send Method): True (<type 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)
```

Success Checksum Error → No message received by client is correct (Content None and Type is <type 'NoneType'>).

Result (Checksum Error -> No message received by client): None (<type 'NoneType'>)

Expectation (Checksum Error -> No message received by client): result = None (<type 'NoneType'>)
 ↪ 'NoneType'>)

A.1.7 An authentication between server and client shall be possible including status feedback methods

Description

The Client shall have a method to initiate the authentication. In case that the server and the client do have identical secrets, the authentication shall be successful.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

Check authentication method feedback (client) and authentication feedback (client and server), in case of differing and identical secrets.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
 ↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
 ↪ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0

prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1

Unittest for socket_protocol

```
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

```

Unittest for socket_protocol

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
```

```
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61  
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Info No secret set

Info Performing Authentication

Success Return Value of authentication method is correct (Content False and Type is <type 'bool'>).

```
Result (Return Value of authentication method): False (<type 'bool'>)
```

```
Expectation (Return Value of authentication method): result = False (<type 'bool'>)
```

Success Authentication state of server is correct (Content True and Type is <type 'bool'>).

```
Result (Authentication state of server): True (<type 'bool'>)
```

Expectation (Authentication state of server): result = True (<type 'bool'>)

Success Authentication state of client is correct (Content True and Type is <type 'bool'>).

Result (Authentication state of client): True (<type 'bool'>)

Expectation (Authentication state of client): result = True (<type 'bool'>)

Info Different secrets set

Success Authentication state of server is correct (Content False and Type is <type 'bool'>).

Result (Authentication state of server): False (<type 'bool'>)

Expectation (Authentication state of server): result = False (<type 'bool'>)

Success Authentication state of client is correct (Content False and Type is <type 'bool'>).

Result (Authentication state of client): False (<type 'bool'>)

Expectation (Authentication state of client): result = False (<type 'bool'>)

Info Performing Authentication

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
 ↪ "None"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
 ↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
 ↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

Unittest for socket_protocol

```

comm-client: TX -> (6): 10 4d cd 55 3a 3e
comm-server: RX <- (6): 10 4d cd 55 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 10 4d cd 55
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
prot-server: Executing callback __authenticate_create_seed__ to process received data
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'66f8120fc584b20713da0eb044b78c7a83fe95c98108855a45bb298a216d2c5b'"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 36 36 66 38 31 32 30 66
↳ 63 35 38 34 62 32 30 37 31 33 64
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 36 36 66 38 31 32 30 66
↳ 63 35 38 34 62 32 30 37 31 33 64
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (64): 61 30 65 62 30 34 34 62 37 38 63 37 61 38 33 66 65 39 35 63 39 38 31
↳ 30 38 38 35 35 61 34 35 62 62 32 39 38 61 32 31 36 64 32 63 35 62 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d c1 41
comm-client: RX <- (64): 61 30 65 62 30 34 34 62 37 38 63 37 61 38 33 66 65 39 35 63 39 38 31
↳ 30 38 38 35 35 61 34 35 62 62 32 39 38 61 32 31 36 64 32 63 35 62 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d c1 41
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (4): 05 2f 3a 3e
comm-client: RX <- (4): 05 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 22 36 36 66 38 31 32 30 66 63 35
↳ 38 34 62 32 30 37 31 33 64 61 30 65 62 30 34 34 62 37 38 63 37 61 38 33 66 65 39 35 63 39
↳ 38 31 30 38 38 35 35 61 34 35 62 62 32 39 38 61 32 31 36 64 32 63 35 62 22 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 20 30 7d c1 41 05 2f
```

```
prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "u'66f8120fc584b20713da0eb044b78c7a83fe95c98108855a45bb298a216d2c5b'"
```

```
prot-client: Executing callback __authenticate_create_key__ to process received data
```

```
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'6d490c66ab570aebec156c39e58f24ee2d2506434466fc6ac8ac35458513c10ce2502956f033a195b7cbcf'
↳ 331bc87ce8bfe3c6e2dec874ff562bb8c406f6658'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 36 64 34 39 30 63 36 36
↳ 61 62 35 37 30 61 65 62 65 65 63
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 36 64 34 39 30 63 36 36
↳ 61 62 35 37 30 61 65 62 65 65 63
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (64): 31 35 36 63 33 39 65 35 38 66 32 34 65 65 32 64 32 35 30 36 34 33 34
↳ 34 36 36 66 63 36 61 63 38 61 63 33 35 34 35 38 35 31 33 63 31 30 63 65 32 35 30 32 39 35
↳ 36 66 30 33 33 61 31 39 35 62 37
```

```
comm-server: RX <- (64): 31 35 36 63 33 39 65 35 38 66 32 34 65 65 32 64 32 35 30 36 34 33 34
↳ 34 36 36 66 63 36 61 63 38 61 63 33 35 34 35 38 35 31 33 63 31 30 63 65 32 35 30 32 39 35
↳ 36 66 30 33 33 61 31 39 35 62 37
```

```
comm-client: TX -> (64): 63 62 63 66 33 33 31 62 63 38 37 63 65 38 62 66 65 33 63 36 65 32 64
↳ 65 63 38 37 34 66 66 35 36 32 62 62 38 63 34 30 36 66 36 36 35 38 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d 2f a1
```

```
comm-server: RX <- (64): 63 62 63 66 33 33 31 62 63 38 37 63 65 38 62 66 65 33 63 36 65 32 64
↳ 65 63 38 37 34 66 66 35 36 32 62 62 38 63 34 30 36 66 36 36 35 38 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d 2f a1
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```



```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (4): 7f c6 3a 3e
comm-server: RX <- (4): 7f c6 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 36 64 34 39 30 63 36 36 61 62
↳ 35 37 30 61 65 62 65 65 63 31 35 36 63 33 39 65 35 38 66 32 34 65 65 32 64 32 35 30 36 34
↳ 33 34 34 36 36 66 63 36 61 63 38 61 63 33 35 34 35 38 35 31 33 63 31 30 63 65 32 35 30 32
↳ 39 35 36 66 30 33 33 61 31 39 35 62 37 63 62 63 66 33 33 31 62 63 38 37 63 65 38 62 66 65
↳ 33 63 36 65 32 64 65 63 38 37 34 66 66 35 36 32 62 62 38 63 34 30 36 66 36 36 35 38 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d 2f a1 7f c6
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "u'6d490c66ab570aebec156c39e58f24ee2d2506434466fc6ac8ac35458513c10ce2502956f033a195b7cbc
↳ f331bc87ce8bfe3c6e2dec874ff562bb8c406f6658'"
prot-server: Executing callback __authenticate_check_key__ to process received data
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "False"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 66 61 6c 73 65 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 31
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 66 61 6c 73 65 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 31
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (7): 7d a1 48 27 7d 3a 3e
comm-client: RX <- (7): 7d a1 48 27 7d 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (63): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 66 61 6c 73 65 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 20 31 7d a1 48 27 7d
```

```
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "False"
```

```
prot-client: Executing callback __authenticate_process_feedback__ to process received data
```

```
prot-client: Got negative authentication feedback
```

Success Return Value of authentication method is correct (Content False and Type is <type 'bool'>).

```
Result (Return Value of authentication method): False (<type 'bool'>)
```

```
Expectation (Return Value of authentication method): result = False (<type 'bool'>)
```

Success Authentication state of server is correct (Content False and Type is <type 'bool'>).

```
Result (Authentication state of server): False (<type 'bool'>)
```

```
Expectation (Authentication state of server): result = False (<type 'bool'>)
```

Success Authentication state of client is correct (Content False and Type is <type 'bool'>).

```
Result (Authentication state of client): False (<type 'bool'>)
```

```
Expectation (Authentication state of client): result = False (<type 'bool'>)
```

Info Identical secrets set

Info Performing Authentication

```
prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 10 4d cd 55 3a 3e
comm-server: RX <- (6): 10 4d cd 55 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 10 4d cd 55
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
prot-server: Executing callback __authenticate_create_seed__ to process received data
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'86c73f84770eaa822f99f920d431e2e5746375067ff8f03fb7f2c533a17635af'"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 38 36 63 37 33 66 38 34
↳ 37 37 30 65 61 61 38 32 32 66 39
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 38 36 63 37 33 66 38 34
↳ 37 37 30 65 61 61 38 32 32 66 39
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (64): 39 66 39 32 30 64 34 33 31 65 32 65 35 37 34 36 33 37 35 30 36 37 66
↳ 66 38 66 30 33 66 62 37 66 32 63 35 33 33 61 31 37 36 33 35 61 66 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 0e 81

```

Unittest for socket_protocol

```
comm-client: RX <- (64): 39 66 39 32 30 64 34 33 31 65 32 65 35 37 34 36 33 37 35 30 36 37 66  
↳ 66 38 66 30 33 66 62 37 66 32 63 35 33 33 61 31 37 36 33 35 61 66 22 2c 20 22 64 61 74 61  
↳ 5f 69 64 22 3a 3d 20 30 7d 0e 81
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (4): 83 02 3a 3e
```

```
comm-client: RX <- (4): 83 02 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 22 38 36 63 37 33 66 38 34 37 37  
↳ 30 65 61 61 38 32 32 66 39 39 66 39 32 30 64 34 33 31 65 32 65 35 37 34 36 33 37 35 30 36  
↳ 37 66 66 38 66 30 33 66 62 37 66 32 63 35 33 33 61 31 37 36 33 35 61 66 22 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 20 30 7d 0e 81 83 02
```

```
prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:  
↳ "u'86c73f84770eaa822f99f920d431e2e5746375067ff8f03fb7f2c533a17635af'"
```

```
prot-client: Executing callback __authenticate_create_key__ to process received data
```

```
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:  
↳ "'20a864dad4b672eeef88ba33b1da828393965f3c3e7cc79a53f1227aacd6c78931b4c8962af810ba32eac7'  
↳ 33df7130776076560d68616bcd01c22388cc4ad75'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 32 30 61 38 36 34 64 61  
↳ 64 62 34 62 36 37 32 65 65 65 66
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 32 30 61 38 36 34 64 61  
↳ 64 62 34 62 36 37 32 65 65 65 66
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (64): 38 38 62 61 33 33 62 31 64 61 38 32 38 33 39 33 39 36 35 66 33 63 33  
↳ 65 37 63 63 37 39 61 35 33 66 31 32 32 37 61 61 63 64 36 63 37 38 39 33 31 62 34 63 38 39  
↳ 36 32 61 66 38 31 30 62 61 33 32
```

Unittest for socket_protocol

```
comm-server: RX <- (64): 38 38 62 61 33 33 62 31 64 61 38 32 38 33 39 33 39 36 35 66 33 63 33
↳ 65 37 63 63 37 39 61 35 33 66 31 32 32 37 61 61 63 64 36 63 37 38 39 33 31 62 34 63 38 39
↳ 36 32 61 66 38 31 30 62 61 33 32
```

```
comm-client: TX -> (64): 65 61 63 37 33 33 64 66 37 31 33 30 37 37 36 30 37 36 35 36 30 64 36
↳ 38 36 31 36 62 63 64 30 31 63 32 32 33 38 38 63 63 34 61 64 37 35 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d b2 17
```

```
comm-server: RX <- (64): 65 61 63 37 33 33 64 66 37 31 33 30 37 37 36 30 37 36 35 36 30 64 36
↳ 38 36 31 36 62 63 64 30 31 63 32 32 33 38 38 63 63 34 61 64 37 35 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d b2 17
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (4): b6 0f 3a 3e
```

```
comm-server: RX <- (4): b6 0f 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 32 30 61 38 36 34 64 61 64 62
↳ 34 62 36 37 32 65 65 65 66 38 38 62 61 33 33 62 31 64 61 38 32 38 33 39 33 39 36 35 66 33
↳ 63 33 65 37 63 63 37 39 61 35 33 66 31 32 32 37 61 61 63 64 36 63 37 38 39 33 31 62 34 63
↳ 38 39 36 32 61 66 38 31 30 62 61 33 32 65 61 63 37 33 33 64 66 37 31 33 30 37 37 36 30 37
↳ 36 35 36 30 64 36 38 36 31 36 62 63 64 30 31 63 32 32 33 38 38 63 63 34 61 64 37 35 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d b2 17 b6 0f
```

```
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "u'20a864dadb4b672eef88ba33b1da828393965f3c3e7cc79a53f1227aacd6c78931b4c8962af810ba32eac
↳ 733df7130776076560d68616bcd01c22388cc4ad75'"
```

```
prot-server: Executing callback __authenticate_check_key__ to process received data
```

```
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 11 d3 26 78 3a 3e
comm-client: RX <- (6): 11 d3 26 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 74 72 75 65 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 31 7d 11 d3 26 78
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
prot-client: Executing callback __authenticate_process_feedback__ to process received data
prot-client: Got positive authentication feedback

```

Success Return Value of authentication method is correct (Content True and Type is <type 'bool'>).

```

Result (Return Value of authentication method): True (<type 'bool'>)
Expectation (Return Value of authentication method): result = True (<type 'bool'>)

```

Success Authentication state of server is correct (Content True and Type is <type 'bool'>).

```

Result (Authentication state of server): True (<type 'bool'>)
Expectation (Authentication state of server): result = True (<type 'bool'>)

```

Success Authentication state of client is correct (Content True and Type is <type 'bool'>).

```

Result (Authentication state of client): True (<type 'bool'>)
Expectation (Authentication state of client): result = True (<type 'bool'>)

```

Info Corrupting the authentication mechanism

```

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

```

Info Performing Authentication

```

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 10 4d cd 55 3a 3e
```

```
comm-server: RX <- (6): 10 4d cd 55 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 10 4d cd 55
```

```
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
```

```
prot-server: Executing callback __authenticate_create_seed__ to process received data
```

Success Return Value of authentication method is correct (Content False and Type is <type 'bool'>).

```
Result (Return Value of authentication method): False (<type 'bool'>)
```

```
Expectation (Return Value of authentication method): result = False (<type 'bool'>)
```

Success Authentication state of server is correct (Content False and Type is <type 'bool'>).

```
Result (Authentication state of server): False (<type 'bool'>)
```

```
Expectation (Authentication state of server): result = False (<type 'bool'>)
```

Success Authentication state of client is correct (Content False and Type is <type 'bool'>).

```
Result (Authentication state of client): False (<type 'bool'>)
```

```
Expectation (Authentication state of client): result = False (<type 'bool'>)
```

A.1.8 An automatic authentication shall available

Description

An authentication is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentication.

Fitcriterion

Check authentication feedback (client and server) after connect has been triggered.

Testresult

This test was passed with the state: **Success**.

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incomming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↳ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↳ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↳ response
prot-server:	Adding Message (service: channel name request, data_id: name) to the ↳ authentication whitelist
prot-server:	Adding Message (service: channel name response, data_id: name) to the ↳ authentication whitelist
prot-server:	Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server:	Adding callback '__channel_name_response__' for SID=9 and DID=0

Unittest for socket_protocol

```
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
```

Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 28 3b d3 54 3a 3e
```

```
comm-server: RX <- (6): 28 3b d3 54 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69  
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61  
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Identical secrets set and automatic authentication

Success Authentication state of server is correct (Content False and Type is <type 'bool'>).

```

Result (Authentication state of server): False (<type 'bool'>)
Expectation (Authentication state of server): result = False (<type 'bool'>)

```

Success Authentication state of client is correct (Content False and Type is <type 'bool'>).

```

Result (Authentication state of client): False (<type 'bool'>)
Expectation (Authentication state of client): result = False (<type 'bool'>)

```

Info Connecting Server and Client

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"

```

Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 28 3b d3 54 3a 3e
```

```
comm-server: RX <- (6): 28 3b d3 54 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: Connection established...
```

```
comm-server: Cleaning up receive-buffer
```

```
prot-server: Cleaning up receive-buffer
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
prot-client: Executing callback __channel_name_response__ to process received data
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

Unittest for socket_protocol

```
comm-client: TX -> (6): 10 4d cd 55 3a 3e
comm-server: RX <- (6): 10 4d cd 55 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 10 4d cd 55
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
prot-server: Executing callback __authenticate_create_seed__ to process received data
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'b00f0c25f2f10414208875b9ac6f524a80f2764f6936bb766019b5c96a867462'"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 62 30 30 66 30 63 32 35
↳ 66 32 66 31 30 34 31 34 32 30 38
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 62 30 30 66 30 63 32 35
↳ 66 32 66 31 30 34 31 34 32 30 38
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (64): 38 37 35 62 39 61 63 36 66 35 32 34 61 38 30 66 32 37 36 34 66 36 39
↳ 33 36 62 62 37 36 36 30 31 39 62 35 63 39 36 61 38 36 37 34 36 32 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 12 06
comm-client: RX <- (64): 38 37 35 62 39 61 63 36 66 35 32 34 61 38 30 66 32 37 36 34 66 36 39
↳ 33 36 62 62 37 36 36 30 31 39 62 35 63 39 36 61 38 36 37 34 36 32 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 12 06
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (4): b7 10 3a 3e
comm-client: RX <- (4): b7 10 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 22 62 30 30 66 30 63 32 35 66 32
↳ 66 31 30 34 31 34 32 30 38 38 37 35 62 39 61 63 36 66 35 32 34 61 38 30 66 32 37 36 34 66
↳ 36 39 33 36 62 62 37 36 36 30 31 39 62 35 63 39 36 61 38 36 37 34 36 32 22 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 20 30 7d 12 06 b7 10
```

```
prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "u'b00f0c25f2f10414208875b9ac6f524a80f2764f6936bb766019b5c96a867462"
```

```
prot-client: Executing callback __authenticate_create_key__ to process received data
```

```
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'2d36bfd165df1f4cce431f464a41856ee33abf7de0eee89466b393958a382e48acdfac5d3ca31194ddf4e1a'
↳ b1ce43ae95834641f28d15dfb8a526734ffa1fa8d'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 32 64 33 36 62 66 64 31
↳ 36 35 64 66 31 66 34 63 63 65 34
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 32 64 33 36 62 66 64 31
↳ 36 35 64 66 31 66 34 63 63 65 34
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (64): 33 31 66 34 36 34 61 34 31 38 35 36 65 65 33 33 61 62 66 37 64 65 30
↳ 65 65 65 38 39 34 36 36 62 33 39 33 39 35 38 61 33 38 32 65 34 38 61 63 64 66 61 63 35 64
↳ 33 63 61 33 31 31 39 34 64 64 66
```

```
comm-server: RX <- (64): 33 31 66 34 36 34 61 34 31 38 35 36 65 65 33 33 61 62 66 37 64 65 30
↳ 65 65 65 38 39 34 36 36 62 33 39 33 39 35 38 61 33 38 32 65 34 38 61 63 64 66 61 63 35 64
↳ 33 63 61 33 31 31 39 34 64 64 66
```

```
comm-client: TX -> (64): 34 65 31 61 62 31 63 65 34 33 61 65 39 35 38 33 34 36 34 31 66 32 38
↳ 64 31 35 64 66 62 38 61 35 32 36 37 33 34 66 66 61 31 66 61 38 64 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d fa d6
```

```
comm-server: RX <- (64): 34 65 31 61 62 31 63 65 34 33 61 65 39 35 38 33 34 36 34 31 66 32 38
↳ 64 31 35 64 66 62 38 61 35 32 36 37 33 34 66 66 61 31 66 61 38 64 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d fa d6
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

Unittest for socket_protocol

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (4): 99 71 3a 3e
comm-server: RX <- (4): 99 71 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 32 64 33 36 62 66 64 31 36 35
↳ 64 66 31 66 34 63 63 65 34 33 31 66 34 36 34 61 34 31 38 35 36 65 65 33 33 61 62 66 37 64
↳ 65 30 65 65 65 38 39 34 36 36 62 33 39 33 39 35 38 61 33 38 32 65 34 38 61 63 64 66 61 63
↳ 35 64 33 63 61 33 31 31 39 34 64 64 66 34 65 31 61 62 31 63 65 34 33 61 65 39 35 38 33 34
↳ 36 34 31 66 32 38 64 31 35 64 66 62 38 61 35 32 36 37 33 34 66 66 61 31 66 61 38 64 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d fa d6 99 71
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "u'2d36bfd165df1f4cce431f464a41856ee33abf7de0eee89466b393958a382e48acdfac5d3ca31194ddf4e1_
↳ ab1ce43ae95834641f28d15dfb8a52673ffa1fa8d'"
prot-server: Executing callback __authenticate_check_key__ to process received data
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 11 d3 26 78 3a 3e
comm-client: RX <- (6): 11 d3 26 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```



```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 74 72 75 65 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 31 7d 11 d3 26 78
```

```
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
```

```
prot-client: Executing callback __authenticate_process_feedback__ to process received data
```

```
prot-client: Got positive authentication feedback
```

Success Authentication state of server is correct (Content True and Type is <type 'bool'>).

```
Result (Authentication state of server): True (<type 'bool'>)
```

```
Expectation (Authentication state of server): result = True (<type 'bool'>)
```

Success Authentication state of client is correct (Content True and Type is <type 'bool'>).

```
Result (Authentication state of client): True (<type 'bool'>)
```

```
Expectation (Authentication state of client): result = True (<type 'bool'>)
```

A.1.9 Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.

Description

Communication (rx and tx) shall be disabled, if a secret is given. Except of a response for registered services, saying that a Authentication is required.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

RX and TX is not possible, till a successfull authentication has been performed.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incomming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
```

```
↳ Response=authentication response
```

Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
```

```

prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Setting a Server secret and no Client secret

Info Transferring a message client → server

```
prot-client: TX -> service: execute request, data_id: 36, status: okay, data:
```

```
↳ "'msg3_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
```

```
↳ 63 65 5f 69 64 22 3a 3d 20 33 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 33 5f 64 61
```

```
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
```

```
↳ 63 65 5f 69 64 22 3a 3d 20 33 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 33 5f 64 61
```

```
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
```

```
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
```

```
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
```

```
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
```

```
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
```

```
↳ 33 36 7d 18 82 9a 08 3a 3e
```

```
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
```

```
↳ 33 36 7d 18 82 9a 08 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
```

```
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
```

```
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
```

```
↳ 63 65 5f 69 64 22 3a 20 33 30 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 33 5f 64 61 74 61
```

```
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
```

```
↳ 20 33 36 7d 18 82 9a 08
```

```
prot-server: RX <- service: execute request, data_id: 36, status: okay, data:
```

```
↳ "u'msg3_data_to_be_transferred'"
```

```
prot-server: Authentication is required. Just sending negative response.
```

```

prot-server: TX -> service: execute response, data_id: 36, status: authentication required,
↳ data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 33 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 33 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 33
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 33 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 33 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 33
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (8): 36 7d 5e 04 41 f5 3a 3e
comm-client: RX <- (8): 36 7d 5e 04 41 f5 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (64): 7b 22 73 74 61 74 75 73 22 3a 20 33 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 33 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 20 33 36 7d 5e 04 41 f5
prot-client: RX <- service: execute response, data_id: 36, status: authentication required,
↳ data: "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content {u'status': 3, u'service_id': 31, u'data': None, u'data_id': 36} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on server side): {u'status': 3, u'service_id': 31, u'data': None, u'data_id': 36} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on server side): result = {'status': 3, 'service_id': 31,
 ↪ 'data': None, 'data_id': 36} (<class 'socket_protocol.data_storage'>)

Info Setting no Server secret but a Client secret

Info Transferring a message server → client

prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
 ↪ "'msg2_data_to_be_transferred'"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
 ↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
 ↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

comm-server: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
 ↪ 33 35 7d 20 18 19 e8 3a 3e

comm-client: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
 ↪ 33 35 7d 20 18 19 e8 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61
 ↪ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
 ↪ 20 33 35 7d 20 18 19 e8

prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
 ↪ "u'msg2_data_to_be_transferred'"

prot-client: Authentication is required. Incomming message will be ignored.

prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↪ in buffer.

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): True (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on client side): None (<type 'NoneType'>)

Expectation (Received message on client side): result = None (<type 'NoneType'>)

Info Identical secrets set

Info Transferring a message client → server

prot-client: Authentication is required. TX-Message service: 17, data_id: 34, status: okay,
↪ data: 'msg1_data_to_be_transfered' will be ignored.

prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↪ in buffer.

Success Returnvalue of Client send Method is correct (Content False and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): False (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = False (<type 'bool'>)

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on server side): None (<type 'NoneType'>)

Expectation (Received message on server side): result = None (<type 'NoneType'>)

Info Transferring a message server → client

prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↪ service or data unknown, data: 'msg2_data_to_be_transfered' will be ignored.

prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↪ in buffer.

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): False (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = False (<type 'bool'>)

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on client side): None (<type 'NoneType'>)

Expectation (Received message on client side): result = None (<type 'NoneType'>)

Info Performing Authentication

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
 ↪ "None"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
 ↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
 ↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 10 4d cd 55 3a 3e

comm-server: RX <- (6): 10 4d cd 55 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
 ↪ 5f 69 64 22 3a 20 30 7d 10 4d cd 55

prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
 ↪ "None"

prot-server: Executing callback __authenticate_create_seed__ to process received data

Unittest for socket_protocol

```

prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'c73c065343bea2e085b402f668f9e146e77f2d00d1b7a1c5dcec31daef3ee95c'"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 63 37 33 63 30 36 35 33
↳ 34 33 62 65 61 32 65 30 38 35 62

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 63 37 33 63 30 36 35 33
↳ 34 33 62 65 61 32 65 30 38 35 62

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (64): 34 30 32 66 36 36 38 66 39 65 31 34 36 65 37 37 66 32 64 30 30 64 31
↳ 62 37 61 31 63 35 64 63 65 63 33 31 64 61 65 66 33 65 65 39 35 63 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 1d 1c

comm-client: RX <- (64): 34 30 32 66 36 36 38 66 39 65 31 34 36 65 37 37 66 32 64 30 30 64 31
↳ 62 37 61 31 63 35 64 63 65 63 33 31 64 61 65 66 33 65 65 39 35 63 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 1d 1c

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (4): 17 cd 3a 3e
comm-client: RX <- (4): 17 cd 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 22 63 37 33 63 30 36 35 33 34 33
↳ 62 65 61 32 65 30 38 35 62 34 30 32 66 36 36 38 66 39 65 31 34 36 65 37 37 66 32 64 30 30
↳ 64 31 62 37 61 31 63 35 64 63 65 63 33 31 64 61 65 66 33 65 65 39 35 63 22 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 20 30 7d 1d 1c 17 cd

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'u'c73c065343bea2e085b402f668f9e146e77f2d00d1b7a1c5dcec31daef3ee95c'"

prot-client: Executing callback __authenticate_create_key__ to process received data

prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'3d396a9e47d89bbef2bc8e5c946b9206c5be702d2bd96c4c91fd22ad1983a3ef2c5ba05139fba3a1c709b97'
↳ a58fafa6c60e6a23a2aecddc73d067387260a76a3'"

```

Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 33 64 33 39 36 61 39 65
↪ 34 37 64 38 39 62 62 65 66 32 62
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 33 64 33 39 36 61 39 65
↪ 34 37 64 38 39 62 62 65 66 32 62
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (64): 63 38 65 35 63 39 34 36 62 39 32 30 36 63 35 62 65 37 30 32 64 32 62
↪ 64 39 36 63 34 63 39 31 66 64 32 32 61 64 31 39 38 33 61 33 65 66 32 63 35 62 61 30 35 31
↪ 33 39 66 62 61 33 61 31 63 37 30
```

```
comm-server: RX <- (64): 63 38 65 35 63 39 34 36 62 39 32 30 36 63 35 62 65 37 30 32 64 32 62
↪ 64 39 36 63 34 63 39 31 66 64 32 32 61 64 31 39 38 33 61 33 65 66 32 63 35 62 61 30 35 31
↪ 33 39 66 62 61 33 61 31 63 37 30
```

```
comm-client: TX -> (64): 39 62 39 37 61 35 38 66 61 66 61 36 63 36 30 65 36 61 32 33 61 32 61
↪ 65 63 64 64 63 37 33 64 30 36 37 33 38 37 32 36 30 61 37 36 61 33 22 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 3d 20 31 7d 40 7e
```

```
comm-server: RX <- (64): 39 62 39 37 61 35 38 66 61 66 61 36 63 36 30 65 36 61 32 33 61 32 61
↪ 65 63 64 64 63 37 33 64 30 36 37 33 38 37 32 36 30 61 37 36 61 33 22 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 3d 20 31 7d 40 7e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (4): 31 8d 3a 3e
```

```
comm-server: RX <- (4): 31 8d 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 33 64 33 39 36 61 39 65 34 37
↪ 64 38 39 62 62 65 66 32 62 63 38 65 35 63 39 34 36 62 39 32 30 36 63 35 62 65 37 30 32 64
↪ 32 62 64 39 36 63 34 63 39 31 66 64 32 32 61 64 31 39 38 33 61 33 65 66 32 63 35 62 61 30
↪ 35 31 33 39 66 62 61 33 61 31 63 37 30 39 62 39 37 61 35 38 66 61 66 61 36 63 36 30 65 36
↪ 61 32 33 61 32 61 65 63 64 64 63 37 33 64 30 36 37 33 38 37 32 36 30 61 37 36 61 33 22 2c
↪ 20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d 40 7e 31 8d
```

Unittest for socket_protocol

```
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "u'3d396a9e47d89bbef2bc8e5c946b9206c5be702d2bd96c4c91fd22ad1983a3ef2c5ba05139fba3a1c709b9
↳ 7a58fafa6c60e6a23a2aecddc73d067387260a76a3'"
prot-server: Executing callback __authenticate_check_key__ to process received data
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 11 d3 26 78 3a 3e
comm-client: RX <- (6): 11 d3 26 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 74 72 75 65 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 31 7d 11 d3 26 78
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
prot-client: Executing callback __authenticate_process_feedback__ to process received data
prot-client: Got positive authentication feedback
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred"
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on server side): {u'status': 0, u'service_id': 17, u'data':

↳ u'msg1_data_to_be_transferred', u'data_id': 34} (<class 'socket_protocol.data_storage'>)

```
Expectation (Received message on server side): result = {'status': 0, 'service_id': 17,
↳ 'data': 'msg1_data_to_be_transferred', 'data_id': 34} (<class
↳ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 35 7d 20 18 19 e8 3a 3e
```

```
comm-client: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 35 7d 20 18 19 e8 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 35 7d 20 18 19 e8
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'u'msg2_data_to_be_transferred'"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): True (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)

Success Received message on client side is correct (Content {'status': 4, 'service_id': 17, 'data': 'msg2_data_to_be_transferred', 'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {'status': 4, 'service_id': 17, 'data':
 ↪ 'msg2_data_to_be_transferred', 'data_id': 35} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on client side): result = {'status': 4, 'service_id': 17,
 ↪ 'data': 'msg2_data_to_be_transferred', 'data_id': 35} (<class
 ↪ 'socket_protocol.data_storage'>)

A.1.10 A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts

Description

It shall be possible to add a specific message, identified by Service-ID and Data-ID, to a whitelist. All messages added to that whitelist shall be transmitted and received, if no authentication was successful performed.

Reason for the implementation

Give the user the possibility to define messages which will not be protected behind the authentication mechanism.

Fitcriterion

Transmission and Reception will be enabled, after the message has been added to the whitelist.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
 ↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
 ↪ authentication whitelist

Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
```



```

prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
```

```
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Identical secrets set

Info Transferring a message client → server

```
prot-client: Authentication is required. TX-Message service: 17, data_id: 34, status: okay,
↳ data: 'msg1_data_to_be_transferred' will be ignored.
```

```
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.
```

Success Returnvalue of Client send Method is correct (Content False and Type is <type 'bool'>).

```
Result (Returnvalue of Client send Method): False (<type 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = False (<type 'bool'>)
```

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

```
Result (Received message on server side): None (<type 'NoneType'>)
```

```
Expectation (Received message on server side): result = None (<type 'NoneType'>)
```

Info Transferring a message server → client

```
prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.
```

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.
```

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

```
Result (Returnvalue of Server send Method): False (<type 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = False (<type 'bool'>)
```

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

```
Result (Received message on client side): None (<type 'NoneType'>)
```

```
Expectation (Received message on client side): result = None (<type 'NoneType'>)
```

Info Added msg1 to client whitelist (sid=17, did=34)

```
prot-client: Adding Message (service: 17, data_id: 34) to the authentication whitelist
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred"
prot-server: Authentication is required. Incomming message will be ignored.
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.

```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on server side): None (<type 'NoneType'>)

Expectation (Received message on server side): result = None (<type 'NoneType'>)

Info Transferring a message server → client

```

prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.

```

```

prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.

```

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): False (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = False (<type 'bool'>)

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on client side): None (<type 'NoneType'>)

Expectation (Received message on client side): result = None (<type 'NoneType'>)

Info Added msg1 to server whitelist (sid=17, did=34)

prot-server: Adding Message (service: 17, data_id: 34) to the authentication whitelist

Info Transferring a message client → server

prot-client: TX -> service: 17, data_id: 34, status: okay, data:

↳ "'msg1_data_to_be_transferred'"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69

↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61

↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69

↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61

↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20

↳ 33 34 7d 7a 6c e4 9b 3a 3e

comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20

↳ 33 34 7d 7a 6c e4 9b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9b
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred"
```

```
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

```
Result (Returnvalue of Client send Method): True (<type 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)
```

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {u'status': 0, u'service_id': 17, u'data':
↳ u'msg1_data_to_be_transferred', u'data_id': 34} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'status': 0, 'service_id': 17,
↳ 'data': 'msg1_data_to_be_transferred', 'data_id': 34} (<class
↳ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.
```

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.
```

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

```
Result (Returnvalue of Server send Method): False (<type 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = False (<type 'bool'>)
```

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

```
Result (Received message on client side): None (<type 'NoneType'>)
```

```
Expectation (Received message on client side): result = None (<type 'NoneType'>)
```

Info Added msg2 to client and server whitelist (sid=17, did=35)

```
prot-client: Adding Message (service: 17, data_id: 35) to the authentication whitelist
```

prot-server: Adding Message (service: 17, data_id: 35) to the authentication whitelist

Info Transferring a message client → server

prot-client: TX -> service: 17, data_id: 34, status: okay, data:

↳ "'msg1_data_to_be_transferred'"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69

↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61

↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69

↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61

↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20

↳ 33 34 7d 7a 6c e4 9b 3a 3e

comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20

↳ 33 34 7d 7a 6c e4 9b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_IDLE

STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69

↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61

↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a

↳ 20 33 34 7d 7a 6c e4 9b

prot-server: RX <- service: 17, data_id: 34, status: okay, data:

↳ "u'msg1_data_to_be_transferred'"

prot-server: Message data is stored in buffer and is now ready to be retrieved by receive

↳ method

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transfered', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on server side): {u'status': 0, u'service_id': 17, u'data':
↪ u'msg1_data_to_be_transfered', u'data_id': 34} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on server side): result = {'status': 0, 'service_id': 17,
↪ 'data': 'msg1_data_to_be_transfered', 'data_id': 34} (<class
↪ 'socket_protocol.data_storage'>)

Info Transferring a message server → client

prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪ "'msg2_data_to_be_transfered'"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-server: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↪ 33 35 7d 20 18 19 e8 3a 3e

comm-client: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↪ 33 35 7d 20 18 19 e8 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA


```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 35 7d 20 18 19 e8
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "u'msg2_data_to_be_transferred"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

```
Result (Returnvalue of Server send Method): True (<type 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)
```

Success Received message on client side is correct (Content {u'status': 4, u'service_id': 17, u'data': u'msg2_data_to_be_transferred', u'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on client side): {u'status': 4, u'service_id': 17, u'data':
```

```
↳ u'msg2_data_to_be_transferred', u'data_id': 35} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on client side): result = {'status': 4, 'service_id': 17,
```

```
↳ 'data': 'msg2_data_to_be_transferred', 'data_id': 35} (<class
```

```
↳ 'socket_protocol.data_storage'>)
```

A.1.11 Define a channel name for the server and client after connection is established

Description

After the connection is established, the client will initiate the channel name exchange. The channel name defined on the client side will be dominant.

Reason for the implementation

Structured logging by creating logger childs for each channel.

Fitcriterion

Perform a channel name exchange with no channel name definition, differing channel name definition and identical channel name definition. In all cases, the channel name of the client will be used. Perform two channel name exchanges with only one channel name definition. This definition will be used.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```

comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

```

```

prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c

```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Setting no Channel name for server and client

Success Channel name of server is correct (Content None and Type is <type 'NoneType'>).

```
Result (Channel name of server): None (<type 'NoneType'>)
Expectation (Channel name of server): result = None (<type 'NoneType'>)
```

Success Channel name of client is correct (Content None and Type is <type 'NoneType'>).

```
Result (Channel name of client): None (<type 'NoneType'>)
Expectation (Channel name of client): result = None (<type 'NoneType'>)
```

Info Setting different Channel names for client and Server

Info Connecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'client'"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 63 6c 69 65 6e 74 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 63 6c 69 65 6e 74 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
comm-client: TX -> (10): 3d 20 30 7d 93 56 e3 b4 3a 3e
comm-server: RX <- (10): 3d 20 30 7d 93 56 e3 b4 3a 3e
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (66): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 22 63 6c 69 65 6e 74 22 2c 20 22
↳ 64 61 74 61 5f 69 64 22 3a 20 30 7d 93 56 e3 b4
prot-server: RX <- service: channel name request, data_id: name, status: okay, data:
↳ "u'client'"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: overwriting user defined channel name from 'server' to u'client'
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Channel name of server is correct (Content 'client' and Type is <type 'str'>).

```

Result (Channel name of server): 'client' (<type 'str'>)
Expectation (Channel name of server): result = 'client' (<type 'str'>)

```

Success Channel name of client is correct (Content 'client' and Type is <type 'str'>).

```

Result (Channel name of client): 'client' (<type 'str'>)
Expectation (Channel name of client): result = 'client' (<type 'str'>)

```

Info Setting identical Channel names for client and server

Info Connecting Server and Client

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'unittest'"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 75 6e 69 74 74 65 73 74
↳ 22 2c 20 22 64 61 74 61 5f 69 64
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 75 6e 69 74 74 65 73 74
↳ 22 2c 20 22 64 61 74 61 5f 69 64
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

```

Unittest for socket_protocol

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (12): 22 3a 3d 20 30 7d b0 bd 92 06 3a 3e
comm-server: RX <- (12): 22 3a 3d 20 30 7d b0 bd 92 06 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (68): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 22 75 6e 69 74 74 65 73 74 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a 20 30 7d b0 bd 92 06
prot-server: RX <- service: channel name request, data_id: name, status: okay, data:
↳ "u'unittest'"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```



```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Channel name of server is correct (Content 'unittest' and Type is <type 'str'>).

```

Result (Channel name of server): 'unittest' (<type 'str'>)
Expectation (Channel name of server): result = 'unittest' (<type 'str'>)

```

Success Channel name of client is correct (Content 'unittest' and Type is <type 'str'>).

```

Result (Channel name of client): 'unittest' (<type 'str'>)
Expectation (Channel name of client): result = 'unittest' (<type 'str'>)

```

Info Setting Channel name for client only

Info Connecting Server and Client

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'client'"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

```

Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 63 6c 69 65 6e 74 22 2c  
↳ 20 22 64 61 74 61 5f 69 64 22 3a
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 63 6c 69 65 6e 74 22 2c  
↳ 20 22 64 61 74 61 5f 69 64 22 3a
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
comm-client: TX -> (10): 3d 20 30 7d 93 56 e3 b4 3a 3e
```

```
comm-server: RX <- (10): 3d 20 30 7d 93 56 e3 b4 3a 3e
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (66): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 22 63 6c 69 65 6e 74 22 2c 20 22  
↳ 64 61 74 61 5f 69 64 22 3a 20 30 7d 93 56 e3 b4
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data:  
↳ "u'client'"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: channel name is now 'client'
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Channel name of server is correct (Content 'client' and Type is <type 'str'>).

```
Result (Channel name of server): 'client' (<type 'str'>)
```

```
Expectation (Channel name of server): result = 'client' (<type 'str'>)
```

Success Channel name of client is correct (Content 'client' and Type is <type 'str'>).

```
Result (Channel name of client): 'client' (<type 'str'>)
```

```
Expectation (Channel name of client): result = 'client' (<type 'str'>)
```

Info Setting Channel name for server only

Info Connecting Server and Client

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer

```

```

prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data:
↳ "'server'"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 22 73 65 72 76 65 72 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 22 73 65 72 76 65 72 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
comm-server: TX -> (10): 3d 20 30 7d 9c 48 3b b3 3a 3e
comm-client: RX <- (10): 3d 20 30 7d 9c 48 3b b3 3a 3e
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (66): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 22 73 65 72 76 65 72 22 2c 20 22
↳ 64 61 74 61 5f 69 64 22 3a 20 30 7d 9c 48 3b b3
prot-client: RX <- service: channel name response, data_id: name, status: okay, data:
↳ "u'server'"
prot-client: Executing callback __channel_name_response__ to process received data
prot-client: channel name is now 'server'

```

Success Channel name of server is correct (Content 'server' and Type is <type 'str'>).

```
Result (Channel name of server): 'server' (<type 'str'>)
```

```
Expectation (Channel name of server): result = 'server' (<type 'str'>)
```

Success Channel name of client is correct (Content 'server' and Type is <type 'str'>).

```
Result (Channel name of client): 'server' (<type 'str'>)
```

```
Expectation (Channel name of client): result = 'server' (<type 'str'>)
```

A.1.12 The User shall be able to define a new service

Description

The service is defined by a Request Service-ID and a Response Service-ID.

Reason for the implementation

Definition of Request and Response SIDs.

Fitcriterion

Define a service and check, that the server will respond on the new Service-ID. The Status shall be "Request has no callback. Data buffered.", because no callback is registered for that request.

Testresult

This test was passed with the state: **Success**.

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incoming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↪ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↪ response
prot-server:	Adding Message (service: channel name request, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding Message (service: channel name response, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server:	Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server:	Adding Service with Request=read data request and Response=read data response
prot-server:	Adding Service with Request=write data request and Response=write data response
prot-server:	Adding Service with Request=execute request and Response=execute response
prot-server:	Initialisation finished.

```

prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```



```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server → client

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred"
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 18; data_id: 34) not
↳ in buffer.
```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

```
Result (Returnvalue of Client send Method): True (<type 'bool'>)
Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)
```

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

```
Result (Received message on server side): None (<type 'NoneType'>)
Expectation (Received message on server side): result = None (<type 'NoneType'>)
```

Info Adding service to server instance for the transmit message

```
prot-server: Adding Service with Request=17 and Response=18
```

Info Transferring a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred"
prot-server: Incoming message with no registered callback. Sending negative response.
prot-server: TX -> service: 18, data_id: 34, status: no callback for service, data buffered,
↳ data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 33
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 33
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (8): 34 7d e8 ee d8 5c 3a 3e
comm-client: RX <- (8): 34 7d e8 ee d8 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (64): 7b 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 20 33 34 7d e8 ee d8 5c
prot-client: RX <- service: 18, data_id: 34, status: no callback for service, data buffered,
↳ data: "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

```

Result (Returnvalue of Client send Method): True (<type 'bool'>)
Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

```

Success Received message on server side is correct (Content {u'status': 1, u'service_id': 18, u'data': None, u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Received message on server side): {u'status': 1, u'service_id': 18, u'data': None,
↳ u'data_id': 34} (<class 'socket_protocol.data_storage'>)
Expectation (Received message on server side): result = {'status': 1, 'service_id': 18,
↳ 'data': None, 'data_id': 34} (<class 'socket_protocol.data_storage'>)

```

A.1.13 Registration of already registered request Service-ID or response Service-ID shall not be possible

Description

An exception shall be raised, if a service registration with an existing request SID or response SID is performed.

Reason for the implementation

Changing existing services will create strange situations with already registered callbacks.

Fitcriterion

Catch exception for registration of existing request and response SID.

Testresult

This test was passed with the state: **Success.**

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

```

```

prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e

```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Adding a service with an already registered request SID

```
prot-server: Service with Request-SID=10 and Response-SID=18 not added, because request SID
↳ is already registered
```

Success Expected Exception RequestSidExistsError was triggered

Info Adding a service with an already registered response SID

```
prot-server: Service with Request-SID=17 and Response-SID=11 not added, because response SID
↳ is already registered
```

Success Expected Exception ResponseSidExistsError was triggered

A.1.14 It shall be possible to register a callback for a specific Service- and Data-ID

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
```


Unittest for socket_protocol

```
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
```

prot-client: Adding Service with Request=execute request and Response=execute response

prot-client: Initialisation finished.

Info Connecting Server and Client

comm-client: Connection established...

comm-client: Cleaning up receive-buffer

prot-client: Cleaning up receive-buffer

prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"

comm-server: Connection established...

comm-server: Cleaning up receive-buffer

prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
 ↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
 ↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 28 3b d3 54 3a 3e

comm-server: RX <- (6): 28 3b d3 54 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
 ↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"

prot-server: Executing callback __channel_name_request__ to process received data

```

prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=10 and DID=0
```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6

```

Unittest for socket_protocol

```

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (5): 17 fc 16 3a 3e

comm-server: RX <- (5): 17 fc 16 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"

prot-server: Executing callback __callback__ to process received data

prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): 02 24 68 3a 3e
comm-client: RX <- (5): 02 24 68 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 33 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 60 02 24 68
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {u'status': 0, u'service.id': 10, u'data': 31, u'data.id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {u'status': 0, u'service.id': 11, u'data': 33, u'data.id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 33,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
↳ 33, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

Info Overwriting existing Callback using one with faulty (too many) return values

```

prot-server: Overwriting existing callback '__callback__' for service_id (10) and data_id (0)
↳ to '__callback_error__'!

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"

```

Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 17 fc 16 3a 3e
```

```
comm-server: RX <- (5): 17 fc 16 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback_error__ to process received data
```

```
prot-server: Exception raised. Check callback __callback_error__: "too many values to unpack"
↳ and it's return values for service: read data request, data_id: 0
```

```
prot-server: TX -> service: read data response, data_id: 0, status: callback error, data:
↳ "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 32 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 30
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 32 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 30
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (7): 7d 3f 8f 7d 86 3a 3e
comm-client: RX <- (7): 7d 3f 8f 7d 86 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (63): 7b 22 73 74 61 74 75 73 22 3a 20 32 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 20 30 7d 3f 8f 7d 86
prot-client: RX <- service: read data response, data_id: 0, status: callback error, data:
↳ "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {u'status': 2, u'service_id': 11, u'data': None, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {u'status': 2, u'service_id': 11, u'data': None,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
Expectation (Message received by client): result = {'status': 2, 'service_id': 11, 'data':
↳ None, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

Info Removing the registered Callback

```

prot-server: Deleting existing callback '__callback_error__' for service_id (10) and data_id
↳ (0)!

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (5): 17 fc 16 3a 3e
comm-server: RX <- (5): 17 fc 16 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↪ 69 64 22 3a 20 30 7d e6 17 fc 16
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Incomming message with no registered callback. Sending negative response.
prot-server: TX -> service: read data response, data_id: 0, status: no callback for service,
↪ data buffered, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↪ 61 74 61 5f 69 64 22 3a 3d 20 30
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↪ 61 74 61 5f 69 64 22 3a 3d 20 30

```



```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (7): 7d 79 5d 48 e2 3a 3e
comm-client: RX <- (7): 7d 79 5d 48 e2 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (63): 7b 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 20 30 7d 79 5d 48 e2
prot-client: RX <- service: read data response, data_id: 0, status: no callback for service,
↳ data buffered, data: "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content None and Type is <type 'NoneType'>).

Result (Message stored inside callback): None (<type 'NoneType'>)

Expectation (Message stored inside callback): result = None (<type 'NoneType'>)

Success Message received by client is correct (Content {u'status': 1, u'service.id': 11, u'data': None, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

Result (Message received by client): {u'status': 1, u'service_id': 11, u'data': None, u'data_id': 0} (<class 'socket_protocol.data_storage'>)

Expectation (Message received by client): result = {'status': 1, 'service_id': 11, 'data': None, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

A.1.15 It shall be possible to register a callback for a specific Service-ID and all Data-IDs**Testresult**

This test was passed with the state: **Success.**

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

```

Unittest for socket_protocol

```
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=10 and DID=None
```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 17 fc 16 3a 3e
comm-server: RX <- (5): 17 fc 16 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

Unittest for socket_protocol

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (5): 02 24 68 3a 3e
comm-client: RX <- (5): 02 24 68 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 33 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 60 02 24 68

prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {u'status': 0, u'service.id': 10, u'data': 31, u'data.id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
 ↪ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0}
 and Type is <class 'socket_protocol.data_storage'>).

Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 33,
 ↪ u'data_id': 0} (<class 'socket_protocol.data_storage'>)

Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
 ↪ 33, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

A.1.16 It shall be possible to register a callback for a specific Data-IDs and all Service-IDs

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
 ↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
 ↪ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0

prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1

prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
 ↪ response

prot-server: Adding Message (service: channel name request, data_id: name) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: channel name response, data_id: name) to the
 ↪ authentication whitelist

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0

prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0

Unittest for socket_protocol

```
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
```


Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 28 3b d3 54 3a 3e
```

```
comm-server: RX <- (6): 28 3b d3 54 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=None and DID=0
```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 17 fc 16 3a 3e
comm-server: RX <- (5): 17 fc 16 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): 02 24 68 3a 3e
comm-client: RX <- (5): 02 24 68 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 33 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 60 02 24 68
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {u'status': 0, u'service.id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {u'status': 0, u'service.id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 33,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
↳ 33, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

A.1.17 It shall be possible to register a callback for all incoming messages

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incoming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
```

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
```

```
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
```

Unittest for socket_protocol

```
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
```

prot-client: Initialisation finished.

Info Connecting Server and Client

comm-client: Connection established...

comm-client: Cleaning up receive-buffer

prot-client: Cleaning up receive-buffer

prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"

comm-server: Connection established...

comm-server: Cleaning up receive-buffer

prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
 ↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
 ↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 28 3b d3 54 3a 3e

comm-server: RX <- (6): 28 3b d3 54 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
 ↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"

prot-server: Executing callback __channel_name_request__ to process received data

prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
```

```
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=None and DID=None
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 17 fc 16 3a 3e
comm-server: RX <- (5): 17 fc 16 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```



```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): 02 24 68 3a 3e
comm-client: RX <- (5): 02 24 68 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 33 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 60 02 24 68
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 33,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
↳ 33, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

A.1.18 Callback choice, if several callbacks are available (caused by wildcard callbacks)

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response

```

Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
```

```

prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering all kind of Callbacks

```

prot-server: Adding callback '__callback3__' for SID=None and DID=None

```

```
prot-server: Adding callback '__callback2__' for SID=None and DID=0
```

```
prot-server: Adding callback '__callback1__' for SID=10 and DID=None
```

```
prot-server: Adding callback '__callback__' for SID=10 and DID=0
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 17 fc 16 3a 3e
```

```
comm-server: RX <- (5): 17 fc 16 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↪ 69 64 22 3a 20 30 7d e6 17 fc 16
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback__ to process received data
```

```
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d 60
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (5): 02 24 68 3a 3e
```

```
comm-client: RX <- (5): 02 24 68 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 33 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 60 02 24 68
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 33,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
↳ 33, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Info Removing Callback for a specific Data- and Service-ID

prot-server: Deleting existing callback '__callback__' for service_id (10) and data_id (0)!

Info Transferring data

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
 ↪ 61 5f 69 64 22 3a 3d 20 30 7d e6

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
 ↪ 61 5f 69 64 22 3a 3d 20 30 7d e6

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

comm-client: TX -> (5): 17 fc 16 3a 3e

comm-server: RX <- (5): 17 fc 16 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
 ↪ 69 64 22 3a 20 30 7d e6 17 fc 16

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"

prot-server: Executing callback __callback1__ to process received data

prot-server: TX -> service: read data response, data_id: 0, status: operation not permitted,
 ↪ data: "34"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 34 2c 20 22 64 61 74
 ↪ 61 5f 69 64 22 3a 3d 20 30 7d 46

Unittest for socket_protocol

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 34 2c 20 22 64 61 74  
↳ 61 5f 69 64 22 3a 3d 20 30 7d 46
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (5): 3f 83 36 3a 3e
```

```
comm-client: RX <- (5): 3f 83 36 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 36 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 34 2c 20 22 64 61 74 61 5f  
↳ 69 64 22 3a 20 30 7d 46 3f 83 36
```

```
prot-client: RX <- service: read data response, data_id: 0, status: operation not permitted,  
↳ data: "34"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive  
↳ method
```

Success Message stored inside callback is correct (Content {u'status': 0, u'service.id': 10, u'data': 31, u'data.id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,  
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,  
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {u'status': 6, u'service.id': 11, u'data': 34, u'data.id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {u'status': 6, u'service_id': 11, u'data': 34,  
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```



```
Expectation (Message received by client): result = {'status': 6, 'service_id': 11, 'data':
↳ 34, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Info Removing Callback for a specific Service-ID and all Data-IDs

```
prot-server: Deleting existing callback '__callback1__' for service_id (10) and data_id
↳ (None)!
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 17 fc 16 3a 3e
```

```
comm-server: RX <- (5): 17 fc 16 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback2__ to process received data
```

```
prot-server: TX -> service: read data response, data_id: 0, status: operation not permitted,
↳ data: "35"
```

Unittest for socket_protocol

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 35 2c 20 22 64 61 74  
↳ 61 5f 69 64 22 3a 3d 20 30 7d e8
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 35 2c 20 22 64 61 74  
↳ 61 5f 69 64 22 3a 3d 20 30 7d e8
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (5): 57 12 a7 3a 3e
```

```
comm-client: RX <- (5): 57 12 a7 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 36 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 35 2c 20 22 64 61 74 61 5f  
↳ 69 64 22 3a 20 30 7d e8 57 12 a7
```

```
prot-client: RX <- service: read data response, data_id: 0, status: operation not permitted,  
↳ data: "35"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive  
↳ method
```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,  
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,  
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {u'status': 6, u'service_id': 11, u'data': 35, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {u'status': 6, u'service_id': 11, u'data': 35,  
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'status': 6, 'service_id': 11, 'data':
↳ 35, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Info Removing Callback for a specific Data-ID and all Serice-IDs

```
prot-server: Deleting existing callback '__callback2__' for service_id (None) and data_id (0)!
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 17 fc 16 3a 3e
```

```
comm-server: RX <- (5): 17 fc 16 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback3__ to process received data
```

```
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "36"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 36 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 1a
```

Unittest for socket_protocol

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 36 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 1a
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (5): 5b f9 7e 3a 3e
```

```
comm-client: RX <- (5): 5b f9 7e 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 36 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 1a 5b f9 7e
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "36"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 36, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 36,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
↳ 36, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

A.1.19 Connection established information

Testresult

This test was passed with the state: **Success.**

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

```

Unittest for socket_protocol

```
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

Unittest for socket_protocol

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Client connection status is correct (Content True and Type is <type 'bool'>).

```

Result (Client connection status): True (<type 'bool'>)
Expectation (Client connection status): result = True (<type 'bool'>)

```

Success Server connection status is correct (Content True and Type is <type 'bool'>).

```

Result (Server connection status): True (<type 'bool'>)
Expectation (Server connection status): result = True (<type 'bool'>)

```

Success Client connection status is correct (Content False and Type is <type 'bool'>).

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
Result (Client connection status): False (<type 'bool'>)
Expectation (Client connection status): result = False (<type 'bool'>)

```

Success Server connection status is correct (Content False and Type is <type 'bool'>).

```

Result (Server connection status): False (<type 'bool'>)
Expectation (Server connection status): result = False (<type 'bool'>)

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

```


Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 28 3b d3 54 3a 3e
```

```
comm-server: RX <- (6): 28 3b d3 54 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61  
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Client connection status is correct (Content True and Type is <type 'bool'>).

```

Result (Client connection status): True (<type 'bool'>)
Expectation (Client connection status): result = True (<type 'bool'>)

```

Success Server connection status is correct (Content True and Type is <type 'bool'>).

```

Result (Server connection status): True (<type 'bool'>)
Expectation (Server connection status): result = True (<type 'bool'>)

```

Info Adding secrets to socket_protocol

Success Client connection status is correct (Content False and Type is <type 'bool'>).

```

Result (Client connection status): False (<type 'bool'>)
Expectation (Client connection status): result = False (<type 'bool'>)

```

Success Server connection status is correct (Content False and Type is <type 'bool'>).

```

Result (Server connection status): False (<type 'bool'>)
Expectation (Server connection status): result = False (<type 'bool'>)

```

Info Doing authentication

```

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"

```

Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 10 4d cd 55 3a 3e
```

```
comm-server: RX <- (6): 10 4d cd 55 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61  
↳ 5f 69 64 22 3a 20 30 7d 10 4d cd 55
```

```
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:  
↳ "None"
```

```
prot-server: Executing callback __authenticate_create_seed__ to process received data
```

```
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:  
↳ "'6674a864f26fb14dc42f9a9df47cc5ce66d879fcb2160c5e026d6e50c9a6b60e'"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 36 36 37 34 61 38 36 34  
↳ 66 32 36 66 62 31 34 64 63 34 32
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 36 36 37 34 61 38 36 34  
↳ 66 32 36 66 62 31 34 64 63 34 32
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (64): 66 39 61 39 64 66 34 37 63 63 35 63 65 36 36 64 38 37 39 66 63 62 32
↳ 31 36 30 63 35 65 30 32 36 64 36 65 35 30 63 39 61 36 62 36 30 65 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 64 96
comm-client: RX <- (64): 66 39 61 39 64 66 34 37 63 63 35 63 65 36 36 64 38 37 39 66 63 62 32
↳ 31 36 30 63 35 65 30 32 36 64 36 65 35 30 63 39 61 36 62 36 30 65 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 64 96
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (4): 3b 4e 3a 3e
comm-client: RX <- (4): 3b 4e 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 22 36 36 37 34 61 38 36 34 66 32
↳ 36 66 62 31 34 64 63 34 32 66 39 61 39 64 66 34 37 63 63 35 63 65 36 36 64 38 37 39 66 63
↳ 62 32 31 36 30 63 35 65 30 32 36 64 36 65 35 30 63 39 61 36 62 36 30 65 22 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 20 30 7d 64 96 3b 4e
prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "u'6674a864f26fb14dc42f9a9df47cc5ce66d879fcb2160c5e026d6e50c9a6b60e'"
prot-client: Executing callback __authenticate_create_key__ to process received data
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'d10b0166f4d91f372e15a9a7248046cad8729e3dae151567e0c84b012f114eb7e8b2bdf2e4091c5267d83fd_
↳ 483dc4ddb46cb5c1eb94b199912cfd37988e8131'"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 64 31 30 62 30 31 36 36
↳ 66 34 64 39 31 66 33 37 32 65 31
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 64 31 30 62 30 31 36 36
↳ 66 34 64 39 31 66 33 37 32 65 31
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (64): 35 61 39 61 37 32 34 38 30 34 36 63 61 64 38 37 32 39 65 33 64 61 65
↳ 31 35 31 35 36 37 65 30 63 38 34 62 30 31 32 66 31 31 34 65 62 37 65 38 62 32 62 64 66 32
↳ 65 34 30 39 31 63 35 32 36 37 64
comm-server: RX <- (64): 35 61 39 61 37 32 34 38 30 34 36 63 61 64 38 37 32 39 65 33 64 61 65
↳ 31 35 31 35 36 37 65 30 63 38 34 62 30 31 32 66 31 31 34 65 62 37 65 38 62 32 62 64 66 32
↳ 65 34 30 39 31 63 35 32 36 37 64
comm-client: TX -> (64): 38 33 66 64 34 38 33 64 63 34 64 64 62 61 34 36 63 62 35 63 31 65 62
↳ 39 34 62 31 39 39 39 31 32 63 66 64 33 37 39 38 38 65 38 31 33 31 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d bb a8
comm-server: RX <- (64): 38 33 66 64 34 38 33 64 63 34 64 64 62 61 34 36 63 62 35 63 31 65 62
↳ 39 34 62 31 39 39 39 31 32 63 66 64 33 37 39 38 38 65 38 31 33 31 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d bb a8
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (4): 5e e7 3a 3e
comm-server: RX <- (4): 5e e7 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 64 31 30 62 30 31 36 36 66 34
↳ 64 39 31 66 33 37 32 65 31 35 61 39 61 37 32 34 38 30 34 36 63 61 64 38 37 32 39 65 33 64
↳ 61 65 31 35 31 35 36 37 65 30 63 38 34 62 30 31 32 66 31 31 34 65 62 37 65 38 62 32 62 64
↳ 66 32 65 34 30 39 31 63 35 32 36 37 64 38 33 66 64 34 38 33 64 63 34 64 64 62 61 34 36 63
↳ 62 35 63 31 65 62 39 34 62 31 39 39 39 31 32 63 66 64 33 37 39 38 38 65 38 31 33 31 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d bb a8 5e e7
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "u'd10b0166f4d91f372e15a9a7248046cad8729e3dae151567e0c84b012f114eb7e8b2bdf2e4091c5267d83f_
↳ d483dc4ddba46cb5c1eb94b199912cfd37988e8131'"
prot-server: Executing callback __authenticate_check_key__ to process received data
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d

```

Unittest for socket_protocol

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61  
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 11 d3 26 78 3a 3e
```

```
comm-client: RX <- (6): 11 d3 26 78 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69  
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 74 72 75 65 2c 20 22 64 61 74 61  
↳ 5f 69 64 22 3a 20 31 7d 11 d3 26 78
```

```
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:  
↳ "True"
```

```
prot-client: Executing callback __authenticate_process_feedback__ to process received data
```

```
prot-client: Got positive authentication feedback
```

Success Client connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Client connection status): True (<type 'bool'>)
```

```
Expectation (Client connection status): result = True (<type 'bool'>)
```

Success Server connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Server connection status): True (<type 'bool'>)
```

```
Expectation (Server connection status): result = True (<type 'bool'>)
```

A.1.20 Is connected information

Testresult

This test was passed with the state: **Success.**

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

```

Unittest for socket_protocol

```
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```


Unittest for socket_protocol

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```

Result (Client Communication instance connection status): True (<type 'bool'>)
Expectation (Client Communication instance connection status): result = True (<type 'bool'>)

```

Success Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```

Result (Server Communication instance connection status): True (<type 'bool'>)
Expectation (Server Communication instance connection status): result = True (<type 'bool'>)

```

Info Disconnecting Server and Client

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

```

Success Client Communication instance connection status is correct (Content False and Type is <type 'bool'>).

```

Result (Client Communication instance connection status): False (<type 'bool'>)
Expectation (Client Communication instance connection status): result = False (<type 'bool'>)

```

Success Server Communication instance connection status is correct (Content False and Type is <type 'bool'>).

```

Result (Server Communication instance connection status): False (<type 'bool'>)
Expectation (Server Communication instance connection status): result = False (<type 'bool'>)

```

A.1.21 Reconnect Method

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```

comm-client: Cleaning up receive-buffer

```

Unittest for socket_protocol

```
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
```

Unittest for socket_protocol

```
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c

```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Client Communication instance connection status): True (<type 'bool'>)
Expectation (Client Communication instance connection status): result = True (<type 'bool'>)
```

Success Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Server Communication instance connection status): True (<type 'bool'>)
Expectation (Server Communication instance connection status): result = True (<type 'bool'>)
```

Info Disconnecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

Success Client Communication instance connection status is correct (Content False and Type is <type 'bool'>).

```
Result (Client Communication instance connection status): False (<type 'bool'>)
Expectation (Client Communication instance connection status): result = False (<type 'bool'>)
```

Success Server Communication instance connection status is correct (Content False and Type is <type 'bool'>).

```
Result (Server Communication instance connection status): False (<type 'bool'>)
Expectation (Server Communication instance connection status): result = False (<type 'bool'>)
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Client Communication instance connection status): True (<type 'bool'>)
Expectation (Client Communication instance connection status): result = True (<type 'bool'>)
```

Success Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Server Communication instance connection status): True (<type 'bool'>)
Expectation (Server Communication instance connection status): result = True (<type 'bool'>)
```

A.1.22 A full Message Object including the defined properties and data shall be transferred.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
```



```

comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1

```

```

prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (21): 3a 3c 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13 3a 3e
comm-server: RX <- (21): 3a 3c 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13 3a 3e
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (17): 00 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (21): 3a 3c 00 00 00 00 00 00 00 09 00 00 00 00 6e 75 6c 6c 12 3a 3e
comm-client: RX <- (21): 3a 3c 00 00 00 00 00 00 00 09 00 00 00 00 6e 75 6c 6c 12 3a 3e
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (17): 00 00 00 00 00 00 00 09 00 00 00 00 6e 75 6c 6c 12
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (45): 3a 3c 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d 3a 3e
```

```
comm-server: RX <- (45): 3a 3c 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (41): 00 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred'"
```

```
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

```
Result (Returnvalue of Client send Method): True (<type 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)
```

Success Received message on server side is correct (Content {'status': 0, 'service_id': 17, 'data': u'msg1_data_to_be_transferred', 'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'status': 0, 'service_id': 17, 'data':
↳ u'msg1_data_to_be_transferred', 'data_id': 34} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'status': 0, 'service_id': 17,
↳ 'data': 'msg1_data_to_be_transferred', 'data_id': 34} (<class
↳ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```

comm-server: TX -> (45): 3a 3c 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61 74
↪ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b 3a 3e

comm-client: RX <- (45): 3a 3c 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61 74
↪ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (41): 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61
↪ 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b

prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↪ "u'msg2_data_to_be_transferred"

prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↪ method

```

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

```
Result (Returnvalue of Server send Method): True (<type 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)
```

Success Received message on client side is correct (Content {'status': 4, 'service_id': 17, 'data': u'msg2_data_to_be_transferred', 'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on client side): {'status': 4, 'service_id': 17, 'data':
↪ u'msg2_data_to_be_transferred', 'data_id': 35} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on client side): result = {'status': 4, 'service_id': 17,
↪ 'data': 'msg2_data_to_be_transferred', 'data_id': 35} (<class
↪ 'socket_protocol.data_storage'>)
```

B Trace for testrun with python 3.8.5 (final)

B.1 Tests with status Info (22)

B.1.1 Status

Description

The Status shall hold some general information (in most cases it is used by the responder). Examples: Okay, Service or Data unknown, Operation not permitted, Authentication required, ...

Reason for the implementation

Give the possibility to transfer additional status information (e.g. to explain negative responses).

Fitcriterion

A Status is part of the Message Object and it is holding the Status information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}

Success status is part of the message object is correct ('status' is in the list or dict).

Result (status is part of the message object): {'data': None, 'data_id': None, 'service_id': None, 'status': None} (<class 'socket_protocol.data_storage'>)

Expectation (status is part of the message object): 'status' in result

Info Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}

Success status is part of the message object is correct ('status' is in the list or dict).

Result (status is part of the message object): {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'} (<class 'socket_protocol.data_storage'>)

Expectation (status is part of the message object): 'status' in result

Success Content in message object for status is correct (Content 'S' and Type is <class 'str'>).

Result (Content in message object for status): 'S' (<class 'str'>)

Expectation (Content in message object for status): result = 'S' (<class 'str'>)

B.1.2 Service-ID

Description

The Service-ID shall hold information about the type of the request / corresponding response. Examples: read request, write request, read response, write response, ...

Reason for the implementation

Give the requestor the possibility to use different types (Services) for a transfer.

Fitcriterion

A Service-ID is part of the Message Object and it is holding the Service-ID information.

Testresult

This test was passed with the state: **Success**.

Info	Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}
-------------	--

Success	service_id is part of the message object is correct ('service_id' is in the list or dict).
----------------	--

Result (service_id is part of the message object):	{'data': None, 'data_id': None, 'service_id': None, 'status': None} (<class 'socket_protocol.data_storage'>)
↳	
Expectation (service_id is part of the message object):	'service_id' in result

Info	Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}
-------------	--

Success	service_id is part of the message object is correct ('service_id' is in the list or dict).
----------------	--

Result (service_id is part of the message object):	{'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'} (<class 'socket_protocol.data_storage'>)
↳	
Expectation (service_id is part of the message object):	'service_id' in result

Success	Content in message object for service_id is correct (Content 'SID' and Type is <class 'str'>).
----------------	--

Result (Content in message object for service_id):	'SID' (<class 'str'>)
Expectation (Content in message object for service_id):	result = 'SID' (<class 'str'>)

B.1.3 Data-ID

Description

The Data-ID shall hold information to differentiate the data for a specific Service.

Reason for the implementation

Give the possibility to transfer different information for each Service.

Fitcriterion

A Data-ID is part of the Message Object and it is holding the Data-ID information.

Testresult

This test was passed with the state: **Success**.

Info	Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}
-------------	--

Success	data_id is part of the message object is correct ('data_id' is in the list or dict).
----------------	--

Result (data_id is part of the message object):	{'data': None, 'data_id': None, 'service_id': None, 'status': None} (<class 'socket_protocol.data_storage'>)
↳	

Expectation (data_id is part of the message object): 'data_id' in result

Info Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}

Success data_id is part of the message object is correct ('data_id' is in the list or dict).

Result (data_id is part of the message object): {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'} (<class 'socket_protocol.data_storage'>)

Expectation (data_id is part of the message object): 'data_id' in result

Success Content in message object for data_id is correct (Content 'DID' and Type is <class 'str'>).

Result (Content in message object for data_id): 'DID' (<class 'str'>)

Expectation (Content in message object for data_id): result = 'DID' (<class 'str'>)

B.1.4 Data

Description

The Data shall hold the data to be transferred. For the most requests not data is transmitted.

Reason for the implementation

Give the possibility to transfer Data.

Fitcriterion

Data is part of the Message Object and it is holding the Data information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}

Success data is part of the message object is correct ('data' is in the list or dict).

Result (data is part of the message object): {'data': None, 'data_id': None, 'service_id': None, 'status': None} (<class 'socket_protocol.data_storage'>)

Expectation (data is part of the message object): 'data' in result

Info Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}

Success data is part of the message object is correct ('data' is in the list or dict).

Result (data is part of the message object): {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'} (<class 'socket_protocol.data_storage'>)

Expectation (data is part of the message object): 'data' in result

Success Content in message object for data is correct (Content 'D' and Type is <class 'str'>).

Result (Content in message object for data): 'D' (<class 'str'>)

Expectation (Content in message object for data): result = 'D' (<class 'str'>)

B.1.5 A full Message Object including the defined properties and data shall be transferred.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incomming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
 ↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
 ↪ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0

prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1

prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
 ↪ response


```

prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer

```

Unittest for socket_protocol

```

prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```

Result (Returnvalue of Client send Method): True (<class 'bool'>)
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)

```

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Received message on server side): {'data_id': 34, 'service_id': 17, 'status': 0,
↳ 'data': 'msg1_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
Expectation (Received message on server side): result = {'service_id': 17, 'data_id': 34,
↳ 'status': 0, 'data': 'msg1_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)

```

Info Transferring a message server → client

```

prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73

```

Unittest for socket_protocol

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
comm-client: RX <- (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 73 e9 96 7f
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): True (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)

Success Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {'data_id': 35, 'service_id': 17, 'status': 4,
↳ 'data': 'msg2_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on client side): result = {'service_id': 17, 'data_id': 35,
↳ 'status': 4, 'data': 'msg2_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)

B.1.6 A checksum shall ensure the correct transmission**Description**

If the checksum does not fit to the checksum of the transferred data, the message will be ignored, because the complete content including the Service- and Data-ID is possibly corrupted.

Reason for the implementation

Ensure correct data transfer.

Fitcriterion

Corrupted message is not in the receive buffer after transmission.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incoming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
```

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
```

```
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
```

```
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
```

```
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
```

```
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
```

```
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

```
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
```

```
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
```

```
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
```

```

prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

```

Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76  
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61  
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76  
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61  
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
```

```
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76  
↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61  
↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76  
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61  
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76  
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61  
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↪ STP_STATE_STORE_DATA
```



```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1c 3a 3e
comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1c
prot-server: Received message has an invalid checksum. Message will be ignored.
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.

```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Checksum Error → No message received by server is correct (Content None and Type is <class 'NoneType'>).

```
Result (Checksum Error -> No message received by server): None (<class 'NoneType'>)
```

```
Expectation (Checksum Error -> No message received by server): result = None (<class
↳ 'NoneType'>)
```

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"

```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73

```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73

```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
comm-client: RX <- (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 73 e9 96 7f
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.

```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): True (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)

Success Checksum Error → No message received by client is correct (Content None and Type is <class 'NoneType'>).

Result (Checksum Error -> No message received by client): None (<class 'NoneType'>)

Expectation (Checksum Error -> No message received by client): result = None (<class 'NoneType'>)

B.1.7 An authentication between server and client shall be possible including status feedback methods

Description

The Client shall have a method to initiate the authentication. In case that the server and the client do have identical secrets, the authentication shall be successfull.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

Check authentication method feedback (client) and authentication feedback (client and server), in case of differing and identical secrets.

Testresult

This test was passed with the state: **Success**.

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incoming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↪ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↪ response
prot-server:	Adding Message (service: channel name request, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding Message (service: channel name response, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server:	Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server:	Adding Service with Request=read data request and Response=read data response
prot-server:	Adding Service with Request=write data request and Response=write data response
prot-server:	Adding Service with Request=execute request and Response=execute response
prot-server:	Initialisation finished.

Unittest for socket_protocol

```
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info No secret set

Info Performing Authentication

Success Return Value of authentication method is correct (Content False and Type is <class 'bool'>).

```

Result (Return Value of authentication method): False (<class 'bool'>)
Expectation (Return Value of authentication method): result = False (<class 'bool'>)

```

Success Authentication state of server is correct (Content True and Type is <class 'bool'>).

```

Result (Authentication state of server): True (<class 'bool'>)
Expectation (Authentication state of server): result = True (<class 'bool'>)

```

Success Authentication state of client is correct (Content True and Type is <class 'bool'>).

```

Result (Authentication state of client): True (<class 'bool'>)
Expectation (Authentication state of client): result = True (<class 'bool'>)

```

Info Different secrets set

Success Authentication state of server is correct (Content False and Type is <class 'bool'>).

```

Result (Authentication state of server): False (<class 'bool'>)
Expectation (Authentication state of server): result = False (<class 'bool'>)

```

Success Authentication state of client is correct (Content False and Type is <class 'bool'>).

Result (Authentication state of client): False (<class 'bool'>)

Expectation (Authentication state of client): result = False (<class 'bool'>)

Info Performing Authentication

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
 ↪ "None"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): fd 82 a2 a9 3a 3e

comm-server: RX <- (6): fd 82 a2 a9 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
 ↪ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9

prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
 ↪ "None"

prot-server: Executing callback __authenticate_create_seed__ to process received data

Unittest for socket_protocol

```

prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'5f0cd9c007043d1f6706919225d48bec8b2bb988c04238f0080253ef30e8e6e1'"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 35 66 30 63

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 35 66 30 63

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (64): 64 39 63 30 30 37 30 34 33 64 31 66 36 37 30 36 39 31 39 32 32 35 64
↳ 34 38 62 65 63 38 62 32 62 62 39 38 38 63 30 34 32 33 38 66 30 30 38 30 32 35 33 65 66 33
↳ 30 65 38 65 36 65 31 22 7d 17 49

comm-client: RX <- (64): 64 39 63 30 30 37 30 34 33 64 31 66 36 37 30 36 39 31 39 32 32 35 64
↳ 34 38 62 65 63 38 62 32 62 62 39 38 38 63 30 34 32 33 38 66 30 30 38 30 32 35 33 65 66 33
↳ 30 65 38 65 36 65 31 22 7d 17 49

comm-server: TX -> (4): a2 77 3a 3e
comm-client: RX <- (4): a2 77 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 35 66 30 63 64 39 63 30 30 37 30 34 33 64 31 66 36 37 30 36 39 31 39 32 32 35
↳ 64 34 38 62 65 63 38 62 32 62 62 39 38 38 63 30 34 32 33 38 66 30 30 38 30 32 35 33 65 66
↳ 33 30 65 38 65 36 65 31 22 7d 17 49 a2 77

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'5f0cd9c007043d1f6706919225d48bec8b2bb988c04238f0080253ef30e8e6e1'"

prot-client: Executing callback __authenticate_create_key__ to process received data

prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'7215bbdbcb13761eaa69d53ab23b1db64a68db7c75a2125207e5696458a49c19c1268696f2208bcdcf513d3j_
↳ eed95b6c7f810b2beeb4711bfb457876a93360ae8'"

```

Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 37 32 31 35
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 37 32 31 35
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (64): 62 62 64 62 63 62 31 33 37 36 31 65 61 61 36 39 64 35 33 61 62 32 33
↳ 62 31 64 62 36 34 61 36 38 64 62 37 63 37 35 61 32 31 32 35 32 30 37 65 35 36 39 36 34 35
↳ 38 61 34 39 63 31 39 63 31 32 36
```

```
comm-server: RX <- (64): 62 62 64 62 63 62 31 33 37 36 31 65 61 61 36 39 64 35 33 61 62 32 33
↳ 62 31 64 62 36 34 61 36 38 64 62 37 63 37 35 61 32 31 32 35 32 30 37 65 35 36 39 36 34 35
↳ 38 61 34 39 63 31 39 63 31 32 36
```

```
comm-client: TX -> (64): 38 36 39 36 66 32 32 30 38 62 63 64 66 63 35 31 33 64 33 65 65 64 39
↳ 35 62 36 63 37 66 38 31 30 62 32 62 65 65 62 34 37 31 31 62 66 62 34 35 37 38 37 36 61 39
↳ 33 33 36 30 61 65 38 22 7d 51 5c
```

```
comm-server: RX <- (64): 38 36 39 36 66 32 32 30 38 62 63 64 66 63 35 31 33 64 33 65 65 64 39
↳ 35 62 36 63 37 66 38 31 30 62 32 62 65 65 62 34 37 31 31 62 66 62 34 35 37 38 37 36 61 39
↳ 33 33 36 30 61 65 38 22 7d 51 5c
```

```
comm-client: TX -> (4): b7 c3 3a 3e
```

```
comm-server: RX <- (4): b7 c3 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 37 32 31 35 62 62 64 62 63 62 31 33 37 36 31 65 61 61 36 39 64 35 33 61 62 32
↳ 33 62 31 64 62 36 34 61 36 38 64 62 37 63 37 35 61 32 31 32 35 32 30 37 65 35 36 39 36 34
↳ 35 38 61 34 39 63 31 39 63 31 32 36 38 36 39 36 66 32 32 30 38 62 63 64 66 63 35 31 33 64
↳ 33 65 65 64 39 35 62 36 63 37 66 38 31 30 62 32 62 65 65 62 34 37 31 31 62 66 62 34 35 37
↳ 38 37 36 61 39 33 33 36 30 61 65 38 22 7d 51 5c b7 c3
```

Unittest for socket_protocol

```
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'7215bbdbcb13761eaa69d53ab23b1db64a68db7c75a2125207e5696458a49c19c1268696f2208bcd513d3_
↳ eed95b6c7f810b2beeb4711bfb457876a93360ae8'"
prot-server: Executing callback __authenticate_check_key__ to process received data
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "False"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 66 61 6c 73 65
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 66 61 6c 73 65
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (7): 7d ea 0a 5c b4 3a 3e
comm-client: RX <- (7): 7d ea 0a 5c b4 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (63): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 66 61 6c 73 65 7d ea 0a 5c b4
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "False"
prot-client: Executing callback __authenticate_process_feedback__ to process received data
prot-client: Got negative authentication feedback
```

Success Return Value of authentication method is correct (Content False and Type is <class 'bool'>).

Result (Return Value of authentication method): False (<class 'bool'>)

Expectation (Return Value of authentication method): result = False (<class 'bool'>)

Success Authentication state of server is correct (Content False and Type is <class 'bool'>).

Result (Authentication state of server): False (<class 'bool'>)

Expectation (Authentication state of server): result = False (<class 'bool'>)

Success Authentication state of client is correct (Content False and Type is <class 'bool'>).

Result (Authentication state of client): False (<class 'bool'>)

Expectation (Authentication state of client): result = False (<class 'bool'>)

Info Identical secrets set

Info Performing Authentication

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
 ↪ "None"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): fd 82 a2 a9 3a 3e

comm-server: RX <- (6): fd 82 a2 a9 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

```

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9

prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"

prot-server: Executing callback __authenticate_create_seed__ to process received data

prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'c7e5a3456599a422b2ca0c711addfe06cf7e063282ae3b6d8a5072883f0c082d'"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 63 37 65 35

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 63 37 65 35

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (64): 61 33 34 35 36 35 39 39 61 34 32 32 62 32 63 61 30 63 37 31 31 61 64
↳ 64 66 65 30 36 63 66 37 65 30 36 33 32 38 32 61 65 33 62 36 64 38 61 35 30 37 32 38 38 33
↳ 66 30 63 30 38 32 64 22 7d be f5

comm-client: RX <- (64): 61 33 34 35 36 35 39 39 61 34 32 32 62 32 63 61 30 63 37 31 31 61 64
↳ 64 66 65 30 36 63 66 37 65 30 36 33 32 38 32 61 65 33 62 36 64 38 61 35 30 37 32 38 38 33
↳ 66 30 63 30 38 32 64 22 7d be f5

comm-server: TX -> (4): 54 ea 3a 3e

comm-client: RX <- (4): 54 ea 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 63 37 65 35 61 33 34 35 36 35 39 39 61 34 32 32 62 32 63 61 30 63 37 31 31 61
↳ 64 64 66 65 30 36 63 66 37 65 30 36 33 32 38 32 61 65 33 62 36 64 38 61 35 30 37 32 38 38
↳ 33 66 30 63 30 38 32 64 22 7d be f5 54 ea

```

Unittest for socket_protocol

```

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'c7e5a3456599a422b2ca0c711adffe06cf7e063282ae3b6d8a5072883f0c082d'"
prot-client: Executing callback __authenticate_create_key__ to process received data
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'5ab03c542a14d3ce74d741329d5c2b61430a40456d806e33cfe2d21d2c264524f7a0eca1da5a5582510dc6d_
↳ b3ccfe7d4377ea2892e41df8ae234bd36d4cf09e1'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 35 61 62 30
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 35 61 62 30
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (64): 33 63 35 34 32 61 31 34 64 33 63 65 37 34 64 37 34 31 33 32 39 64 35
↳ 63 32 62 36 31 34 33 30 61 34 30 34 35 36 64 38 30 36 65 33 33 63 66 65 32 64 32 31 64 32
↳ 63 32 36 34 35 32 34 66 37 61 30
comm-server: RX <- (64): 33 63 35 34 32 61 31 34 64 33 63 65 37 34 64 37 34 31 33 32 39 64 35
↳ 63 32 62 36 31 34 33 30 61 34 30 34 35 36 64 38 30 36 65 33 33 63 66 65 32 64 32 31 64 32
↳ 63 32 36 34 35 32 34 66 37 61 30
comm-client: TX -> (64): 65 63 61 31 64 61 35 61 35 35 38 32 35 31 30 64 63 36 64 62 33 63 63
↳ 66 65 37 64 34 33 37 37 65 61 32 38 39 32 65 34 31 64 66 38 61 65 32 33 34 62 64 33 36 64
↳ 34 63 66 30 39 65 31 22 7d 4f c4
comm-server: RX <- (64): 65 63 61 31 64 61 35 61 35 35 38 32 35 31 30 64 63 36 64 62 33 63 63
↳ 66 65 37 64 34 33 37 37 65 61 32 38 39 32 65 34 31 64 66 38 61 65 32 33 34 62 64 33 36 64
↳ 34 63 66 30 39 65 31 22 7d 4f c4
comm-client: TX -> (4): 6f 98 3a 3e
comm-server: RX <- (4): 6f 98 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

Unittest for socket_protocol

```
STP: message identified - (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 35 61 62 30 33 63 35 34 32 61 31 34 64 33 63 65 37 34 64 37 34 31 33 32 39 64
↳ 35 63 32 62 36 31 34 33 30 61 34 30 34 35 36 64 38 30 36 65 33 33 63 66 65 32 64 32 31 64
↳ 32 63 32 36 34 35 32 34 66 37 61 30 65 63 61 31 64 61 35 61 35 35 38 32 35 31 30 64 63 36
↳ 64 62 33 63 63 66 65 37 64 34 33 37 37 65 61 32 38 39 32 65 34 31 64 66 38 61 65 32 33 34
↳ 62 64 33 36 64 34 63 66 30 39 65 31 22 7d 4f c4 6f 98
```

```
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'5ab03c542a14d3ce74d741329d5c2b61430a40456d806e33cfe2d21d2c264524f7a0eca1da5a5582510dc6d_
↳ b3ccfe7d4377ea2892e41df8ae234bd36d4cf09e1'"
```

```
prot-server: Executing callback __authenticate_check_key__ to process received data
```

```
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 94 fe 74 32 3a 3e
```

```
comm-client: RX <- (6): 94 fe 74 32 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 74 72 75 65 7d 94 fe 74 32
```

```
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
```

```
prot-client: Executing callback __authenticate_process_feedback__ to process received data
prot-client: Got positive authentication feedback
```

Success Return Value of authentication method is correct (Content True and Type is <class 'bool'>).

```
Result (Return Value of authentication method): True (<class 'bool'>)
Expectation (Return Value of authentication method): result = True (<class 'bool'>)
```

Success Authentication state of server is correct (Content True and Type is <class 'bool'>).

```
Result (Authentication state of server): True (<class 'bool'>)
Expectation (Authentication state of server): result = True (<class 'bool'>)
```

Success Authentication state of client is correct (Content True and Type is <class 'bool'>).

```
Result (Authentication state of client): True (<class 'bool'>)
Expectation (Authentication state of client): result = True (<class 'bool'>)
```

Info Corrupting the authentication mechanism

```
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

Info Performing Authentication

```
prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```



```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
prot-server: Executing callback __authenticate_create_seed__ to process received data

```

Success Return Value of authentication method is correct (Content False and Type is <class 'bool'>).

```
Result (Return Value of authentication method): False (<class 'bool'>)
```

```
Expectation (Return Value of authentication method): result = False (<class 'bool'>)
```

Success Authentication state of server is correct (Content False and Type is <class 'bool'>).

```
Result (Authentication state of server): False (<class 'bool'>)
```

```
Expectation (Authentication state of server): result = False (<class 'bool'>)
```

Success Authentication state of client is correct (Content False and Type is <class 'bool'>).

```
Result (Authentication state of client): False (<class 'bool'>)
```

```
Expectation (Authentication state of client): result = False (<class 'bool'>)
```

B.1.8 An automatic authentication shall available

Description

An authentication is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentication.

Fitcriterion

Check authentication feedback (client and server) after connect has been triggered.

Testresult

This test was passed with the state: **Success.**

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

```

Unittest for socket_protocol

```
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e

```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Identical secrets set and automatic authentication

Success Authentication state of server is correct (Content False and Type is <class 'bool'>).

```
Result (Authentication state of server): False (<class 'bool'>)
Expectation (Authentication state of server): result = False (<class 'bool'>)
```

Success Authentication state of client is correct (Content False and Type is <class 'bool'>).

```
Result (Authentication state of client): False (<class 'bool'>)
Expectation (Authentication state of client): result = False (<class 'bool'>)
```

Info Connecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

Unittest for socket_protocol

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

Unittest for socket_protocol

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
prot-server: Executing callback __authenticate_create_seed__ to process received data
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'3a3237ba0428bcf48ee3dcfb249aa00dcec1449cf44a09053fab8f26414a3814'"
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

Unittest for socket_protocol

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 33 61 33 32
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 33 61 33 32
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (64): 33 37 62 61 30 34 32 38 62 63 66 34 38 65 65 33 64 63 66 62 32 34 39
↳ 61 61 30 30 64 63 65 63 31 34 34 39 63 66 34 34 61 30 39 30 35 33 66 61 62 38 66 32 36 34
↳ 31 34 61 33 38 31 34 22 7d aa d5
```

```
comm-client: RX <- (64): 33 37 62 61 30 34 32 38 62 63 66 34 38 65 65 33 64 63 66 62 32 34 39
↳ 61 61 30 30 64 63 65 63 31 34 34 39 63 66 34 34 61 30 39 30 35 33 66 61 62 38 66 32 36 34
↳ 31 34 61 33 38 31 34 22 7d aa d5
```

```
comm-server: TX -> (4): 6a 51 3a 3e
```

```
comm-client: RX <- (4): 6a 51 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 33 61 33 32 33 37 62 61 30 34 32 38 62 63 66 34 38 65 65 33 64 63 66 62 32 34
↳ 39 61 61 30 30 64 63 65 63 31 34 34 39 63 66 34 34 61 30 39 30 35 33 66 61 62 38 66 32 36
↳ 34 31 34 61 33 38 31 34 22 7d aa d5 6a 51
```

```
prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'3a3237ba0428bcf48ee3dcfb249aa00dcec1449cf44a09053fab8f26414a3814'"
```



```

prot-client: Executing callback __authenticate_create_key__ to process received data
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'8ad8a387c28241b38382d2f4646e10f85588c6ea360f393e7960022d028ef599155e9cb81186bb5cb4a63fb'
↳ b57105f83ba1917d6beffb4d0549c521c4e1dd549'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 38 61 64 38
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 38 61 64 38
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (64): 61 33 38 37 63 32 38 32 34 31 62 33 38 33 38 32 64 32 66 34 36 34 36
↳ 65 31 30 66 38 35 35 38 38 63 36 65 61 33 36 30 66 33 39 33 65 37 39 36 30 30 32 32 64 30
↳ 32 38 65 66 35 39 39 31 35 35 65
comm-server: RX <- (64): 61 33 38 37 63 32 38 32 34 31 62 33 38 33 38 32 64 32 66 34 36 34 36
↳ 65 31 30 66 38 35 35 38 38 63 36 65 61 33 36 30 66 33 39 33 65 37 39 36 30 30 32 32 64 30
↳ 32 38 65 66 35 39 39 31 35 35 65
comm-client: TX -> (64): 39 63 62 38 31 31 38 36 62 62 35 63 62 34 61 36 33 66 62 62 35 37 31
↳ 30 35 66 38 33 62 61 31 39 31 37 64 36 62 65 66 66 62 34 64 30 35 34 39 63 35 32 31 63 34
↳ 65 31 64 64 35 34 39 22 7d 40 75
comm-server: RX <- (64): 39 63 62 38 31 31 38 36 62 62 35 63 62 34 61 36 33 66 62 62 35 37 31
↳ 30 35 66 38 33 62 61 31 39 31 37 64 36 62 65 66 66 62 34 64 30 35 34 39 63 35 32 31 63 34
↳ 65 31 64 64 35 34 39 22 7d 40 75
comm-client: TX -> (4): 98 4a 3a 3e
comm-server: RX <- (4): 98 4a 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```
STP: message identified - (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 38 61 64 38 61 33 38 37 63 32 38 32 34 31 62 33 38 33 38 32 64 32 66 34 36 34
↳ 36 65 31 30 66 38 35 35 38 38 63 36 65 61 33 36 30 66 33 39 33 65 37 39 36 30 30 32 32 64
↳ 30 32 38 65 66 35 39 39 31 35 35 65 39 63 62 38 31 31 38 36 62 62 35 63 62 34 61 36 33 66
↳ 62 62 35 37 31 30 35 66 38 33 62 61 31 39 31 37 64 36 62 65 66 66 62 34 64 30 35 34 39 63
↳ 35 32 31 63 34 65 31 64 64 35 34 39 22 7d 40 75 98 4a
```

```
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'8ad8a387c28241b38382d2f4646e10f85588c6ea360f393e7960022d028ef599155e9cb81186bb5cb4a63fb_'
↳ b57105f83ba1917d6beffb4d0549c521c4e1dd549'"
```

```
prot-server: Executing callback __authenticate_check_key__ to process received data
```

```
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 94 fe 74 32 3a 3e
```

```
comm-client: RX <- (6): 94 fe 74 32 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 74 72 75 65 7d 94 fe 74 32
```

```
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
```

```
prot-client: Executing callback __authenticate_process_feedback__ to process received data
```

```
prot-client: Got positive authentication feedback
```

Success Authentication state of server is correct (Content True and Type is <class 'bool'>).

```
Result (Authentication state of server): True (<class 'bool'>)
```

```
Expectation (Authentication state of server): result = True (<class 'bool'>)
```

Success Authentication state of client is correct (Content True and Type is <class 'bool'>).

```
Result (Authentication state of client): True (<class 'bool'>)
```

```
Expectation (Authentication state of client): result = True (<class 'bool'>)
```

B.1.9 Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.

Description

Communication (rx and tx) shall be disabled, if a secret is given. Except of a response for registered services, saying that a Authentication is required.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

RX and TX is not possible, till a successfull authentication has been performed.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incomming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
```

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
```

Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
```

```
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
```

```

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Setting a Server secret and no Client secret

Info Transferring a message client → server

```

prot-client: TX -> service: execute request, data_id: 36, status: okay, data:
↳ "'msg3_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 36 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 33 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73

```

Unittest for socket_protocol

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 36 2c 20 22 73 65 72  
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 33 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22  
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (32): 67 33 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65  
↳ 64 22 7d 13 e9 64 3d 3a 3e
```

```
comm-server: RX <- (32): 67 33 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65  
↳ 64 22 7d 13 e9 64 3d 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 36 2c 20 22 73 65 72  
↳ 76 69 63 65 5f 69 64 22 3a 20 33 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61  
↳ 74 61 22 3a 20 22 6d 73 67 33 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72  
↳ 65 64 22 7d 13 e9 64 3d
```

```
prot-server: RX <- service: execute request, data_id: 36, status: okay, data:  
↳ "'msg3_data_to_be_transferred'"
```

```
prot-server: Authentication is required. Just sending negative response.
```

```
prot-server: TX -> service: execute response, data_id: 36, status: authentication required,  
↳ data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 36 2c 20 22 73 65 72  
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 33 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 33 2c 20 22  
↳ 64 61 74 61 22 3a 3d 20 6e 75 6c
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 36 2c 20 22 73 65 72  
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 33 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 33 2c 20 22  
↳ 64 61 74 61 22 3a 3d 20 6e 75 6c
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (8): 6c 7d 5d 78 af a4 3a 3e
comm-client: RX <- (8): 6c 7d 5d 78 af a4 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (64): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 36 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 33 31 2c 20 22 73 74 61 74 75 73 22 3a 20 33 2c 20 22 64 61
↳ 74 61 22 3a 20 6e 75 6c 6c 7d 5d 78 af a4
prot-client: RX <- service: execute response, data_id: 36, status: authentication required,
↳ data: "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content {'data_id': 36, 'service_id': 31, 'status': 3, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'data_id': 36, 'service_id': 31, 'status': 3,
↳ 'data': None} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'service_id': 31, 'data_id': 36,
↳ 'status': 3, 'data': None} (<class 'socket_protocol.data_storage'>)
```

Info Setting no Server secret but a Client secret

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```


Unittest for socket_protocol

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72  
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22  
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72  
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22  
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65  
↳ 64 22 7d 73 e9 96 7f 3a 3e
```

```
comm-client: RX <- (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65  
↳ 64 22 7d 73 e9 96 7f 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 35 2c 20 22 73 65 72  
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 64 61  
↳ 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72  
↳ 65 64 22 7d 73 e9 96 7f
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:  
↳ "'msg2_data_to_be_transferred'"
```

```
prot-client: Authentication is required. Incomming message will be ignored.
```

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not  
↳ in buffer.
```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Server send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)
```

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

Result (Received message on client side): None (<class 'NoneType'>)

Expectation (Received message on client side): result = None (<class 'NoneType'>)

Info Identical secrets set

Info Transferring a message client → server

prot-client: Authentication is required. TX-Message service: 17, data_id: 34, status: okay,
↳ data: 'msg1_data_to_be_transferred' will be ignored.

prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.

Success Returnvalue of Client send Method is correct (Content False and Type is <class 'bool'>).

Result (Returnvalue of Client send Method): False (<class 'bool'>)

Expectation (Returnvalue of Client send Method): result = False (<class 'bool'>)

Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).

Result (Received message on server side): None (<class 'NoneType'>)

Expectation (Received message on server side): result = None (<class 'NoneType'>)

Info Transferring a message server → client

prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.

prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.

Success Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): False (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = False (<class 'bool'>)

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

Result (Received message on client side): None (<class 'NoneType'>)

Expectation (Received message on client side): result = None (<class 'NoneType'>)

Info Performing Authentication

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"

Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76  
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61  
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76  
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61  
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
```

```
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76  
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61  
↳ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9
```

```
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:  
↳ "None"
```

```
prot-server: Executing callback __authenticate_create_seed__ to process received data
```

```
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:  
↳ "'e724ac5d8c8afc57a6f7d39c60c2994d1770cfd9de3e49a38d2d2626d43510c'"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76  
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61  
↳ 74 61 22 3a 3d 20 22 65 37 32 34
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76  
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61  
↳ 74 61 22 3a 3d 20 22 65 37 32 34
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (64): 61 63 35 64 38 63 38 61 66 63 35 37 61 36 66 37 64 33 39 63 36 30 63
↳ 32 39 39 34 64 31 37 37 30 63 66 63 64 39 64 65 33 65 34 39 61 33 38 64 32 64 32 36 32 36
↳ 64 34 33 35 31 30 63 22 7d f7 f1

comm-client: RX <- (64): 61 63 35 64 38 63 38 61 66 63 35 37 61 36 66 37 64 33 39 63 36 30 63
↳ 32 39 39 34 64 31 37 37 30 63 66 63 64 39 64 65 33 65 34 39 61 33 38 64 32 64 32 36 32 36
↳ 64 34 33 35 31 30 63 22 7d f7 f1

comm-server: TX -> (4): d9 c5 3a 3e
comm-client: RX <- (4): d9 c5 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 65 37 32 34 61 63 35 64 38 63 38 61 66 63 35 37 61 36 66 37 64 33 39 63 36 30
↳ 63 32 39 39 34 64 31 37 37 30 63 66 63 64 39 64 65 33 65 34 39 61 33 38 64 32 64 32 36 32
↳ 36 64 34 33 35 31 30 63 22 7d f7 f1 d9 c5

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'e724ac5d8c8afc57a6f7d39c60c2994d1770cfdc9de3e49a38d2d2626d43510c'"

prot-client: Executing callback __authenticate_create_key__ to process received data
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'53d050f59d7ddd4caf6c2e530df0a70730a4d7411f62d8b2ffc40cc7e5ba530dc0d994e42cb191fb2254c92'
↳ 1c1e449633a9183df456ae84ea4fbdea588fefcf5'"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 35 33 64 30

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 35 33 64 30

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

Unittest for socket_protocol

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (64): 35 30 66 35 39 64 37 64 64 64 34 63 61 66 36 63 32 65 35 33 30 64 66
↳ 30 61 37 30 37 33 30 61 34 64 37 34 31 31 66 36 32 64 38 62 32 66 66 63 34 30 63 63 37 65
↳ 35 62 61 35 33 30 64 63 30 64 39
comm-server: RX <- (64): 35 30 66 35 39 64 37 64 64 64 34 63 61 66 36 63 32 65 35 33 30 64 66
↳ 30 61 37 30 37 33 30 61 34 64 37 34 31 31 66 36 32 64 38 62 32 66 66 63 34 30 63 63 37 65
↳ 35 62 61 35 33 30 64 63 30 64 39
comm-client: TX -> (64): 39 34 65 34 32 63 62 31 39 31 66 62 32 32 35 34 63 39 32 31 63 31 65
↳ 34 34 39 36 33 33 61 39 31 38 33 64 66 34 35 36 61 65 38 34 65 61 34 66 62 64 65 61 35 38
↳ 38 66 65 66 63 66 35 22 7d 98 3f
comm-server: RX <- (64): 39 34 65 34 32 63 62 31 39 31 66 62 32 32 35 34 63 39 32 31 63 31 65
↳ 34 34 39 36 33 33 61 39 31 38 33 64 66 34 35 36 61 65 38 34 65 61 34 66 62 64 65 61 35 38
↳ 38 66 65 66 63 66 35 22 7d 98 3f
comm-client: TX -> (4): 25 6b 3a 3e
comm-server: RX <- (4): 25 6b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 35 33 64 30 35 30 66 35 39 64 37 64 64 64 34 63 61 66 36 63 32 65 35 33 30 64
↳ 66 30 61 37 30 37 33 30 61 34 64 37 34 31 31 66 36 32 64 38 62 32 66 66 63 34 30 63 63 37
↳ 65 35 62 61 35 33 30 64 63 30 64 39 39 34 65 34 32 63 62 31 39 31 66 62 32 32 35 34 63 39
↳ 32 31 63 31 65 34 34 39 36 33 33 61 39 31 38 33 64 66 34 35 36 61 65 38 34 65 61 34 66 62
↳ 64 65 61 35 38 38 66 65 66 63 66 35 22 7d 98 3f 25 6b
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'53d050f59d7ddd4caf6c2e530df0a70730a4d7411f62d8b2fffc40cc7e5ba530dc0d994e42cb191fb2254c92_'
↳ '1c1e449633a9183df456ae84ea4fbdea588fefcf5'"
prot-server: Executing callback __authenticate_check_key__ to process received data
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d

```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 94 fe 74 32 3a 3e
```

```
comm-client: RX <- (6): 94 fe 74 32 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 74 72 75 65 7d 94 fe 74 32
```

```
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
```

```
prot-client: Executing callback __authenticate_process_feedback__ to process received data
```

```
prot-client: Got positive authentication feedback
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```

Result (Returnvalue of Client send Method): True (<class 'bool'>)
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)

```

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).

```

Result (Received message on server side): {'data_id': 34, 'service_id': 17, 'status': 0,
↳ 'data': 'msg1_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
Expectation (Received message on server side): result = {'service_id': 17, 'data_id': 34,
↳ 'status': 0, 'data': 'msg1_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)

```

Info Transferring a message server → client

```

prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"

```

Unittest for socket_protocol

```

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
comm-client: RX <- (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 73 e9 96 7f
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): True (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)

Success Received message on client side is correct (Content {'data.id': 35, 'service.id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {'data_id': 35, 'service_id': 17, 'status': 4,
↳ 'data': 'msg2_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)


```
Expectation (Received message on client side): result = {'service_id': 17, 'data_id': 35,
↳ 'status': 4, 'data': 'msg2_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)
```

B.1.10 A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts

Description

It shall be possible to add a specific message, identified by Service-ID and Data-ID, to a whitelist. All messages added to that whitelist shall be transmitted and received, if no authentication was successful performed.

Reason for the implementation

Give the user the possibility to define messages which will not be protected behind the authentication mechanism.

Fitcriterion

Transmission and Reception will be enabled, after the message has been added to the whitelist.

Testresult

This test was passed with the state: **Success**.

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incoming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↳ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↳ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↳ response

```

prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer

```

Unittest for socket_protocol

```

prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Identical secrets set

Info Transferring a message client → server

```

prot-client: Authentication is required. TX-Message service: 17, data_id: 34, status: okay,
↳ data: 'msg1_data_to_be_transferred' will be ignored.
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.

```

Success Returnvalue of Client send Method is correct (Content False and Type is <class 'bool'>).

```

Result (Returnvalue of Client send Method): False (<class 'bool'>)
Expectation (Returnvalue of Client send Method): result = False (<class 'bool'>)

```

Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).

```

Result (Received message on server side): None (<class 'NoneType'>)
Expectation (Received message on server side): result = None (<class 'NoneType'>)

```

Info Transferring a message server → client

```
prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.
```

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.
```

Success Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).

```
Result (Returnvalue of Server send Method): False (<class 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = False (<class 'bool'>)
```

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

```
Result (Received message on client side): None (<class 'NoneType'>)
```

```
Expectation (Received message on client side): result = None (<class 'NoneType'>)
```

Info Added msg1 to client whitelist (sid=17, did=34)

```
prot-client: Adding Message (service: 17, data_id: 34) to the authentication whitelist
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
prot-server: Authentication is required. Incomming message will be ignored.
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.

```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```

Result (Returnvalue of Client send Method): True (<class 'bool'>)
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)

```

Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).

```

Result (Received message on server side): None (<class 'NoneType'>)
Expectation (Received message on server side): result = None (<class 'NoneType'>)

```

Info Transferring a message server → client

```

prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.

```

Success Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).

```

Result (Returnvalue of Server send Method): False (<class 'bool'>)
Expectation (Returnvalue of Server send Method): result = False (<class 'bool'>)

```

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

```

Result (Received message on client side): None (<class 'NoneType'>)

```

Expectation (Received message on client side): result = None (<class 'NoneType'>)

Info Added msg1 to server whitelist (sid=17, did=34)

prot-server: Adding Message (service: 17, data_id: 34) to the authentication whitelist

Info Transferring a message client → server

prot-client: TX -> service: 17, data_id: 34, status: okay, data:

↳ "'msg1_data_to_be_transferred'"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72

↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22

↳ 64 61 74 61 22 3a 3d 20 22 6d 73

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72

↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22

↳ 64 61 74 61 22 3a 3d 20 22 6d 73

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65

↳ 64 22 7d 4c bc bd 1b 3a 3e

comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65

↳ 64 22 7d 4c bc bd 1b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_IDLE

STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72

↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61

↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72

↳ 65 64 22 7d 4c bc bd 1b

prot-server: RX <- service: 17, data_id: 34, status: okay, data:

↳ "'msg1_data_to_be_transferred'"

```
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'data_id': 34, 'service_id': 17, 'status': 0,
↳ 'data': 'msg1_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'service_id': 17, 'data_id': 34,
↳ 'status': 0, 'data': 'msg1_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.
```

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.
```

Success Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).

```
Result (Returnvalue of Server send Method): False (<class 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = False (<class 'bool'>)
```

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

```
Result (Received message on client side): None (<class 'NoneType'>)
```

```
Expectation (Received message on client side): result = None (<class 'NoneType'>)
```

Info Added msg2 to client and server whitelist (sid=17, did=35)

```
prot-client: Adding Message (service: 17, data_id: 35) to the authentication whitelist
```

```
prot-server: Adding Message (service: 17, data_id: 35) to the authentication whitelist
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```


Unittest for socket_protocol

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72  
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22  
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65  
↳ 64 22 7d 4c bc bd 1b 3a 3e
```

```
comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65  
↳ 64 22 7d 4c bc bd 1b 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72  
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61  
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72  
↳ 65 64 22 7d 4c bc bd 1b
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:  
↳ "'msg1_data_to_be_transferred'"
```

```
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive  
↳ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'data_id': 34, 'service_id': 17, 'status': 0,  
↳ 'data': 'msg1_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'service_id': 17, 'data_id': 34,
↳ 'status': 0, 'data': 'msg1_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
```

```
comm-client: RX <- (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 73 e9 96 7f
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): True (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)

Success Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {'data_id': 35, 'service_id': 17, 'status': 4,
↪ 'data': 'msg2_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on client side): result = {'service_id': 17, 'data_id': 35,
↪ 'status': 4, 'data': 'msg2_data_to_be_transferred'} (<class
↪ 'socket_protocol.data_storage'>)

B.1.11 Define a channel name for the server and client after connection is established

Description

After the connection is established, the client will initiate the channel name exchange. The channel name defined on the client side will be dominant.

Reason for the implementation

Structured logging by creating logger childs for each channel.

Fitcriterion

Perform a channel name exchange with no channel name definition, differing channel name definition and identical channel name definition. In all cases, the channel name of the client will be used. Perform two channel name exchanges with only one channel name definition. This definition will be used.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incomming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↪ authentication whitelist

Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
```

```

prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

```

```

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Setting no Channel name for server and client

Success Channel name of server is correct (Content None and Type is <class 'NoneType'>).

Result (Channel name of server): None (<class 'NoneType'>)

Expectation (Channel name of server): result = None (<class 'NoneType'>)

Success Channel name of client is correct (Content None and Type is <class 'NoneType'>).

Result (Channel name of client): None (<class 'NoneType'>)

Expectation (Channel name of client): result = None (<class 'NoneType'>)

Info Setting different Channel names for client and Server

Info Connecting Server and Client

comm-client: Connection Lost...

prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-server: Connection Lost...

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-client: Connection established...

comm-client: Cleaning up receive-buffer

prot-client: Cleaning up receive-buffer

prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↪ "'client'"

comm-server: Connection established...

comm-server: Cleaning up receive-buffer

prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 22 63 6c 69 65

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 22 63 6c 69 65

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (10): 6e 74 22 7d ee af 7b 7e 3a 3e
comm-server: RX <- (10): 6e 74 22 7d ee af 7b 7e 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (66): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 63 6c 69 65 6e 74 22 7d ee af 7b 7e
prot-server: RX <- service: channel name request, data_id: name, status: okay, data:
↳ "'client'"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: overwriting user defined channel name from 'server' to 'client'
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f

```



```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Channel name of server is correct (Content 'client' and Type is <class 'str'>).

```
Result (Channel name of server): 'client' (<class 'str'>)
Expectation (Channel name of server): result = 'client' (<class 'str'>)
```

Success Channel name of client is correct (Content 'client' and Type is <class 'str'>).

```
Result (Channel name of client): 'client' (<class 'str'>)
Expectation (Channel name of client): result = 'client' (<class 'str'>)
```

Info Setting identical Channel names for client and server

Info Connecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'unittest'"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 75 6e 69 74
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 75 6e 69 74
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

Unittest for socket_protocol

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (12): 74 65 73 74 22 7d f8 f6 c9 e9 3a 3e
comm-server: RX <- (12): 74 65 73 74 22 7d f8 f6 c9 e9 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (68): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 75 6e 69 74 74 65 73 74 22 7d f8 f6 c9 e9
prot-server: RX <- service: channel name request, data_id: name, status: okay, data:
↳ "'unittest'"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Channel name of server is correct (Content 'unittest' and Type is <class 'str'>).

```
Result (Channel name of server): 'unittest' (<class 'str'>)
```

```
Expectation (Channel name of server): result = 'unittest' (<class 'str'>)
```

Success Channel name of client is correct (Content 'unittest' and Type is <class 'str'>).

```
Result (Channel name of client): 'unittest' (<class 'str'>)
```

```
Expectation (Channel name of client): result = 'unittest' (<class 'str'>)
```

Info Setting Channel name for client only

Info Connecting Server and Client

```
comm-client: Connection Lost...
```

```
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

```
comm-server: Connection Lost...
```

```
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

```
comm-client: Connection established...
```

```
comm-client: Cleaning up receive-buffer
```

```
prot-client: Cleaning up receive-buffer
```

```
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'client'"
```

```
comm-server: Connection established...
```

```
comm-server: Cleaning up receive-buffer
```

```
prot-server: Cleaning up receive-buffer
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 63 6c 69 65
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 63 6c 69 65
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

Unittest for socket_protocol

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (10): 6e 74 22 7d ee af 7b 7e 3a 3e
comm-server: RX <- (10): 6e 74 22 7d ee af 7b 7e 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (66): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 63 6c 69 65 6e 74 22 7d ee af 7b 7e
prot-server: RX <- service: channel name request, data_id: name, status: okay, data:
↳ "'client'"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: channel name is now 'client'
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Channel name of server is correct (Content 'client' and Type is <class 'str'>).

```

Result (Channel name of server): 'client' (<class 'str'>)
Expectation (Channel name of server): result = 'client' (<class 'str'>)

```

Success Channel name of client is correct (Content 'client' and Type is <class 'str'>).

```

Result (Channel name of client): 'client' (<class 'str'>)
Expectation (Channel name of client): result = 'client' (<class 'str'>)

```

Info Setting Channel name for server only

Info Connecting Server and Client

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data:
↳ "'server'"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 73 65 72 76
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 73 65 72 76
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (10): 65 72 22 7d ac a3 7b cc 3a 3e
comm-client: RX <- (10): 65 72 22 7d ac a3 7b cc 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (66): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 73 65 72 76 65 72 22 7d ac a3 7b cc
prot-client: RX <- service: channel name response, data_id: name, status: okay, data:
↳ "'server'"
prot-client: Executing callback __channel_name_response__ to process received data
prot-client: channel name is now 'server'

```

Success Channel name of server is correct (Content 'server' and Type is <class 'str'>).

```

Result (Channel name of server): 'server' (<class 'str'>)
Expectation (Channel name of server): result = 'server' (<class 'str'>)

```

Success Channel name of client is correct (Content 'server' and Type is <class 'str'>).

```

Result (Channel name of client): 'server' (<class 'str'>)
Expectation (Channel name of client): result = 'server' (<class 'str'>)

```

B.1.12 The User shall be able to define a new service

Description

The service is defined by a Request Service-ID and a Response Service-ID.

Reason for the implementation

Definition of Request and Response SIDs.

Fitcriterion

Define a service and check, that the server will respond on the new Service-ID. The Status shall be "Request has no callback. Data buffered.", because no callback is registered for that request.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
```


Unittest for socket_protocol

```
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e

```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Transferring a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1b
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 18; data_id: 34) not
↳ in buffer.
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).

```
Result (Received message on server side): None (<class 'NoneType'>)
```

```
Expectation (Received message on server side): result = None (<class 'NoneType'>)
```

Info Adding service to server instance for the transmit message

```
prot-server: Adding Service with Request=17 and Response=18
```

Info Transferring a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
prot-server: Incoming message with no registered callback. Sending negative response.
prot-server: TX -> service: 18, data_id: 34, status: no callback for service, data buffered,
↳ data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 6e 75 6c
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 6e 75 6c
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (8): 6c 7d bd 30 46 9b 3a 3e
comm-client: RX <- (8): 6c 7d bd 30 46 9b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```
STP: message identified - (64): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 38 2c 20 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 64 61
↳ 74 61 22 3a 20 6e 75 6c 6c 7d bd 30 46 9b
```

```
prot-client: RX <- service: 18, data_id: 34, status: no callback for service, data buffered,
↳ data: "None"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 18, 'status': 1, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'data_id': 34, 'service_id': 18, 'status': 1,
↳ 'data': None} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'service_id': 18, 'data_id': 34,
↳ 'status': 1, 'data': None} (<class 'socket_protocol.data_storage'>)
```

B.1.13 Registration of already registered request Service-ID or response Service-ID shall not be possible

Description

An exception shall be raised, if a service registration with an existing request SID or response SID is performed.

Reason for the implementation

Changing existing services will create strange situations with already registered callbacks.

Fitcriterion

Catch exception for registration of existing request and response SID.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incoming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
```

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
```

Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
```

```

prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```


STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
 ↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"

prot-server: Executing callback __channel_name_request__ to process received data

prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

comm-server: TX -> (6): 30 59 be 2f 3a 3e

comm-client: RX <- (6): 30 59 be 2f 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
 ↪ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f

prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"

prot-client: Executing callback __channel_name_response__ to process received data

Info Adding a service with an already registered request SID

prot-server: Service with Request-SID=10 and Response-SID=18 not added, because request SID
 ↪ is already registered

Success Expected Exception RequestSidExistsError was triggered

Info Adding a service with an already registered response SID

```
prot-server: Service with Request-SID=17 and Response-SID=11 not added, because response SID
↳ is already registered
```

Success Expected Exception ResponseSidExistsError was triggered

B.1.14 It shall be possible to register a callback for a specific Service- and Data-ID

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
```

Unittest for socket_protocol

```
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
```

```

prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=10 and DID=0
```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

Unittest for socket_protocol

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 33 7d e4 e1 8c bb
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':
↳ 33} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':
↳ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Info Overwriting existing Callback using one with faulty (too many) return values

```
prot-server: Overwriting existing callback '__callback__' for service_id (10) and data_id (0)
↳ to '__callback_error__'!
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

Unittest for socket_protocol

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback_error__ to process received data
prot-server: Exception raised. Check callback __callback_error__: "too many values to unpack
↳ (expected 2)" and it's return values for service: read data request, data_id: 0
prot-server: TX -> service: read data response, data_id: 0, status: callback error, data:
↳ "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 32 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 6e 75 6c 6c
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 32 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 6e 75 6c 6c
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (7): 7d a1 a2 87 f3 3a 3e
comm-client: RX <- (7): 7d a1 a2 87 f3 3a 3e

```


STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (63): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 32 2c 20 22 64 61 74
 ↪ 61 22 3a 20 6e 75 6c 6c 7d a1 a2 87 f3

prot-client: RX <- service: read data response, data_id: 0, status: callback error, data:
 ↪ "None"

prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
 ↪ method

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
 ↪ 'data': 31} (<class 'socket_protocol.data_storage'>)

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
 ↪ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 2, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 2, 'data':
 ↪ None} (<class 'socket_protocol.data_storage'>)

Expectation (Message received by client): result = {'data': None, 'data_id': 0, 'service_id':
 ↪ 11, 'status': 2} (<class 'socket_protocol.data_storage'>)

Info Removing the registered Callback

prot-server: Deleting existing callback '__callback_error__' for service_id (10) and data_id
 ↪ (0)!

Info Transferring data

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
 ↪ 61 74 61 22 3a 3d 20 33 31 7d b8

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
 ↪ 61 74 61 22 3a 3d 20 33 31 7d b8

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Incomming message with no registered callback. Sending negative response.
prot-server: TX -> service: read data response, data_id: 0, status: no callback for service,
↳ data buffered, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 6e 75 6c 6c
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 6e 75 6c 6c
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (7): 7d 88 6a 33 01 3a 3e
comm-client: RX <- (7): 7d 88 6a 33 01 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (63): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 64 61 74
↳ 61 22 3a 20 6e 75 6c 6c 7d 88 6a 33 01
prot-client: RX <- service: read data response, data_id: 0, status: no callback for service,
↳ data buffered, data: "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content None and Type is <class 'NoneType'>).

```

Result (Message stored inside callback): None (<class 'NoneType'>)
Expectation (Message stored inside callback): result = None (<class 'NoneType'>)

```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 1, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 1, 'data':
↳ None} (<class 'socket_protocol.data_storage'>)
Expectation (Message received by client): result = {'data': None, 'data_id': 0, 'service_id':
↳ 11, 'status': 1} (<class 'socket_protocol.data_storage'>)

```

B.1.15 It shall be possible to register a callback for a specific Service-ID and all Data-IDs

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

```

Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
```

```

prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 30 59 be 2f 3a 3e
```

```
comm-client: RX <- (6): 30 59 be 2f 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=10 and DID=None
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 5b f5 78 3a 3e
```

```
comm-server: RX <- (5): 5b f5 78 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↪ 61 22 3a 20 33 31 7d b8 5b f5 78
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback__ to process received data
```

```
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 33 7d e4
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 33 7d e4
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 33 7d e4 e1 8c bb
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':
↳ 33} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':
↳ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

B.1.16 It shall be possible to register a callback for a specific Data-IDs and all Service-IDs

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```

comm-client: Cleaning up receive-buffer

```


Unittest for socket_protocol

```
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
```

```

prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f

```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=None and DID=0
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 5b f5 78 3a 3e
```

```
comm-server: RX <- (5): 5b f5 78 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback__ to process received data
```

```
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4
```

Unittest for socket_protocol

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76  
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64  
↳ 61 74 61 22 3a 3d 20 33 33 7d e4
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (5): e1 8c bb 3a 3e
```

```
comm-client: RX <- (5): e1 8c bb 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->  
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76  
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74  
↳ 61 22 3a 20 33 33 7d e4 e1 8c bb
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive  
↳ method
```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,  
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,  
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':  
↳ 33} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':  
↳ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

B.1.17 It shall be possible to register a callback for all incoming messages

Testresult

This test was passed with the state: **Success.**

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
    
```

```

prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```



```

comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=None and DID=None
```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

Unittest for socket_protocol

```
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 33 7d e4 e1 8c bb

prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)
```

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
 ↪ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and
 Type is <class 'socket_protocol.data_storage'>).

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':
 ↪ 33} (<class 'socket_protocol.data_storage'>)

Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':
 ↪ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)

B.1.18 Callback choice, if several callbacks are available (caused by wildcard callbacks)

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
 ↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
 ↪ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0

prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1

prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
 ↪ response

prot-server: Adding Message (service: channel name request, data_id: name) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: channel name response, data_id: name) to the
 ↪ authentication whitelist

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0

Unittest for socket_protocol

```
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
```

Unittest for socket_protocol

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
```

```
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering all kind of Callbacks

```

prot-server: Adding callback '__callback3__' for SID=None and DID=None
prot-server: Adding callback '__callback2__' for SID=None and DID=0
prot-server: Adding callback '__callback1__' for SID=10 and DID=None
prot-server: Adding callback '__callback__' for SID=10 and DID=0

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 33 7d e4 e1 8c bb
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':
↳ 33} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':
↳ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Info Removing Callback for a specific Data- and Service-ID

```
prot-server: Deleting existing callback '__callback__' for service_id (10) and data_id (0)!
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```


Unittest for socket_protocol

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback1__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: operation not permitted,
↳ data: "34"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 34 7d 53
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 34 7d 53
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): 62 51 ca 3a 3e
comm-client: RX <- (5): 62 51 ca 3a 3e
```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 36 2c 20 22 64 61 74
 ↪ 61 22 3a 20 33 34 7d 53 62 51 ca

prot-client: RX <- service: read data response, data_id: 0, status: operation not permitted,
 ↪ data: "34"

prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
 ↪ method

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
 ↪ 'data': 31} (<class 'socket_protocol.data_storage'>)

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
 ↪ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 6, 'data': 34} and Type is <class 'socket_protocol.data_storage'>).

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 6, 'data':
 ↪ 34} (<class 'socket_protocol.data_storage'>)

Expectation (Message received by client): result = {'data': 34, 'data_id': 0, 'service_id':
 ↪ 11, 'status': 6} (<class 'socket_protocol.data_storage'>)

Info Removing Callback for a specific Service-ID and all Data-IDs

prot-server: Deleting existing callback '__callback1__' for service_id (10) and data_id
 ↪ (None)!

Info Transferring data

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
 ↪ 61 74 61 22 3a 3d 20 33 31 7d b8

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
 ↪ 61 74 61 22 3a 3d 20 33 31 7d b8

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback2__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: operation not permitted,
↳ data: "35"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 35 7d 4a
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 35 7d 4a
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): 79 60 8b 3a 3e
comm-client: RX <- (5): 79 60 8b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 36 2c 20 22 64 61 74
↳ 61 22 3a 20 33 35 7d 4a 79 60 8b
prot-client: RX <- service: read data response, data_id: 0, status: operation not permitted,
↳ data: "35"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)
Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 6, 'data': 35} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 6, 'data':
↳ 35} (<class 'socket_protocol.data_storage'>)
Expectation (Message received by client): result = {'data': 35, 'data_id': 0, 'service_id':
↳ 11, 'status': 6} (<class 'socket_protocol.data_storage'>)
```

Info Removing Callback for a specific Data-ID and all Serice-IDs

```
prot-server: Deleting existing callback '__callback2__' for service_id (None) and data_id (0)!
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

Unittest for socket_protocol

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback3__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "36"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 36 7d 99
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 36 7d 99
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): 96 78 fe 3a 3e
comm-client: RX <- (5): 96 78 fe 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 36 7d 99 96 78 fe
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "36"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 36} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':
↳ 36} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message received by client): result = {'data': 36, 'data_id': 0, 'service_id':
↳ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

B.1.19 Connection established information

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```

comm-client: Cleaning up receive-buffer

```

```

comm-server: Cleaning up receive-buffer

```

```

comm-server: Waiting for incoming connection

```

```

prot-server: Cleaning up receive-buffer

```

```

prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response

```

```

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

```

Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
```

```

prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```



```

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Client connection status is correct (Content True and Type is <class 'bool'>).

Result (Client connection status): True (<class 'bool'>)

Expectation (Client connection status): result = True (<class 'bool'>)

Success Server connection status is correct (Content True and Type is <class 'bool'>).

Result (Server connection status): True (<class 'bool'>)

Expectation (Server connection status): result = True (<class 'bool'>)

Success Client connection status is correct (Content False and Type is <class 'bool'>).

comm-client: Connection Lost...

prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-server: Connection Lost...

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

Result (Client connection status): False (<class 'bool'>)

Expectation (Client connection status): result = False (<class 'bool'>)

Success Server connection status is correct (Content False and Type is <class 'bool'>).

Result (Server connection status): False (<class 'bool'>)

Expectation (Server connection status): result = False (<class 'bool'>)

Info Connecting Server and Client

comm-client: Connection established...

comm-client: Cleaning up receive-buffer

prot-client: Cleaning up receive-buffer

prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"

comm-server: Connection established...

comm-server: Cleaning up receive-buffer

prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Client connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Client connection status): True (<class 'bool'>)
```

```
Expectation (Client connection status): result = True (<class 'bool'>)
```

Success Server connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Server connection status): True (<class 'bool'>)
```

```
Expectation (Server connection status): result = True (<class 'bool'>)
```

Info Adding secrets to socket_protocol

Success Client connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Client connection status): False (<class 'bool'>)
```

```
Expectation (Client connection status): result = False (<class 'bool'>)
```

Success Server connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Server connection status): False (<class 'bool'>)
```

```
Expectation (Server connection status): result = False (<class 'bool'>)
```

Info Doing authentication

```
prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
prot-server: Executing callback __authenticate_create_seed__ to process received data
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'9bb48ec8638270557aaffe5d80e4088c626d78c335256480b66fdb6daf505ea'"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 39 62 62 34
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 39 62 62 34
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

Unittest for socket_protocol

```
comm-server: TX -> (64): 38 65 63 38 36 33 38 32 37 30 35 35 37 61 61 66 66 65 35 64 38 30 65
↳ 34 30 38 38 63 36 32 36 64 37 38 63 33 33 35 32 35 36 34 38 30 62 36 36 66 64 62 63 36 64
↳ 61 66 35 30 35 65 61 22 7d 4d 30
```

```
comm-client: RX <- (64): 38 65 63 38 36 33 38 32 37 30 35 35 37 61 61 66 66 65 35 64 38 30 65
↳ 34 30 38 38 63 36 32 36 64 37 38 63 33 33 35 32 35 36 34 38 30 62 36 36 66 64 62 63 36 64
↳ 61 66 35 30 35 65 61 22 7d 4d 30
```

```
comm-server: TX -> (4): 43 80 3a 3e
```

```
comm-client: RX <- (4): 43 80 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 39 62 62 34 38 65 63 38 36 33 38 32 37 30 35 35 37 61 61 66 66 65 35 64 38 30
↳ 65 34 30 38 38 63 36 32 36 64 37 38 63 33 33 35 32 35 36 34 38 30 62 36 36 66 64 62 63 36
↳ 64 61 66 35 30 35 65 61 22 7d 4d 30 43 80
```

```
prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'9bb48ec8638270557aaffe5d80e4088c626d78c335256480b66fdb6daf505ea'"
```

```
prot-client: Executing callback __authenticate_create_key__ to process received data
```

```
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'1896e78394ec5309ef0184c803d96539357900ce28428e9ee33b090f938776a696c4e8f9fbae78d6a02d873'
↳ 69441b147b562161e3df3092303835d3bb372b784'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 31 38 39 36
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 31 38 39 36
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

Unittest for socket_protocol

```
comm-client: TX -> (64): 65 37 38 33 39 34 65 63 35 33 30 39 65 66 30 31 38 34 63 38 30 33 64
↳ 39 36 35 33 39 33 35 37 39 30 30 63 65 32 38 34 32 38 65 39 65 65 33 33 62 30 39 30 66 39
↳ 33 38 37 37 36 61 36 39 36 63 34
```

```
comm-server: RX <- (64): 65 37 38 33 39 34 65 63 35 33 30 39 65 66 30 31 38 34 63 38 30 33 64
↳ 39 36 35 33 39 33 35 37 39 30 30 63 65 32 38 34 32 38 65 39 65 65 33 33 62 30 39 30 66 39
↳ 33 38 37 37 36 61 36 39 36 63 34
```

```
comm-client: TX -> (64): 65 38 66 39 66 62 61 65 37 38 64 36 61 30 32 64 38 37 33 36 39 34 34
↳ 31 62 31 34 37 62 35 36 32 31 36 31 65 33 64 66 33 30 39 32 33 30 33 38 33 35 64 33 62 62
↳ 33 37 32 62 37 38 34 22 7d 75 e6
```

```
comm-server: RX <- (64): 65 38 66 39 66 62 61 65 37 38 64 36 61 30 32 64 38 37 33 36 39 34 34
↳ 31 62 31 34 37 62 35 36 32 31 36 31 65 33 64 66 33 30 39 32 33 30 33 38 33 35 64 33 62 62
↳ 33 37 32 62 37 38 34 22 7d 75 e6
```

```
comm-client: TX -> (4): 88 37 3a 3e
```

```
comm-server: RX <- (4): 88 37 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 31 38 39 36 65 37 38 33 39 34 65 63 35 33 30 39 65 66 30 31 38 34 63 38 30 33
↳ 64 39 36 35 33 39 33 35 37 39 30 30 63 65 32 38 34 32 38 65 39 65 65 33 33 62 30 39 30 66
↳ 39 33 38 37 37 36 61 36 39 36 63 34 65 38 66 39 66 62 61 65 37 38 64 36 61 30 32 64 38 37
↳ 33 36 39 34 34 31 62 31 34 37 62 35 36 32 31 36 31 65 33 64 66 33 30 39 32 33 30 33 38 33
↳ 35 64 33 62 62 33 37 32 62 37 38 34 22 7d 75 e6 88 37
```

```
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'1896e78394ec5309ef0184c803d96539357900ce28428e9ee33b090f938776a696c4e8f9fbae78d6a02d873'
↳ 69441b147b562161e3df3092303835d3bb372b784'"
```

```
prot-server: Executing callback __authenticate_check_key__ to process received data
```

```
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 94 fe 74 32 3a 3e
comm-client: RX <- (6): 94 fe 74 32 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 74 72 75 65 7d 94 fe 74 32
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
prot-client: Executing callback __authenticate_process_feedback__ to process received data
prot-client: Got positive authentication feedback

```

Success Client connection status is correct (Content True and Type is <class 'bool'>).

```

Result (Client connection status): True (<class 'bool'>)
Expectation (Client connection status): result = True (<class 'bool'>)

```

Success Server connection status is correct (Content True and Type is <class 'bool'>).

```

Result (Server connection status): True (<class 'bool'>)
Expectation (Server connection status): result = True (<class 'bool'>)

```

B.1.20 Is connected information

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

```


Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
```

```

prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).

Result (Client Communication instance connection status): True (<class 'bool'>)

Expectation (Client Communication instance connection status): result = True (<class 'bool'>)

Success Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).

Result (Server Communication instance connection status): True (<class 'bool'>)

Expectation (Server Communication instance connection status): result = True (<class 'bool'>)

Info Disconnecting Server and Client

comm-client: Connection Lost...

prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-server: Connection Lost...

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

Success Client Communication instance connection status is correct (Content False and Type is <class 'bool'>).

Result (Client Communication instance connection status): False (<class 'bool'>)

Expectation (Client Communication instance connection status): result = False (<class 'bool'>)

Success Server Communication instance connection status is correct (Content False and Type is <class 'bool'>).

Result (Server Communication instance connection status): False (<class 'bool'>)

Expectation (Server Communication instance connection status): result = False (<class 'bool'>)

B.1.21 Reconnect Method

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
 ↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
 ↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
 ↪ authentication whitelist

Unittest for socket_protocol

```
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
```

prot-client: Adding Service with Request=execute request and Response=execute response

prot-client: Initialisation finished.

Info Connecting Server and Client

comm-client: Connection established...

comm-client: Cleaning up receive-buffer

prot-client: Cleaning up receive-buffer

prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"

comm-server: Connection established...

comm-server: Cleaning up receive-buffer

prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 53 5e 67 0b 3a 3e

comm-server: RX <- (6): 53 5e 67 0b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
 ↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"

prot-server: Executing callback __channel_name_request__ to process received data

```

prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).

Result (Client Communication instance connection status): True (<class 'bool'>)

Expectation (Client Communication instance connection status): result = True (<class 'bool'>)

Success Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).

Result (Server Communication instance connection status): True (<class 'bool'>)

Expectation (Server Communication instance connection status): result = True (<class 'bool'>)

Info Disconnecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

Success Client Communication instance connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Client Communication instance connection status): False (<class 'bool'>)
Expectation (Client Communication instance connection status): result = False (<class 'bool'>)
```

Success Server Communication instance connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Server Communication instance connection status): False (<class 'bool'>)
Expectation (Server Communication instance connection status): result = False (<class 'bool'>)
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
prot-server: Cleaning up receive-buffer
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```



```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Client Communication instance connection status): True (<class 'bool'>)
```

```
Expectation (Client Communication instance connection status): result = True (<class 'bool'>)
```

Success Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Server Communication instance connection status): True (<class 'bool'>)
```

```
Expectation (Server Communication instance connection status): result = True (<class 'bool'>)
```

B.1.22 A full Message Object including the defined properties and data shall be transfered.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incomming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
```

```
↪ Response=authentication response
```

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
```

```
↪ authentication whitelist
```

```
prot-server: Adding Message (service: authentication response, data_id: seed) to the
```

```
↪ authentication whitelist
```

```
prot-server: Adding Message (service: authentication request, data_id: key) to the
```

```
↪ authentication whitelist
```

Unittest for socket_protocol

```
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
```

```
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (21): 3a 3c 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13 3a 3e
comm-server: RX <- (21): 3a 3c 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13 3a 3e
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (17): 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (21): 3a 3c 00 00 00 00 00 00 09 00 00 00 00 6e 75 6c 6c 12 3a 3e
comm-client: RX <- (21): 3a 3c 00 00 00 00 00 00 09 00 00 00 00 6e 75 6c 6c 12 3a 3e
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (17): 00 00 00 00 00 00 00 09 00 00 00 00 6e 75 6c 6c 12
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (45): 3a 3c 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d 3a 3e
```

```
comm-server: RX <- (45): 3a 3c 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (41): 00 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'data_id': 34, 'service_id': 17, 'status': 0,
↳ 'data': 'msg1_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'service_id': 17, 'data_id': 34,
↳ 'status': 0, 'data': 'msg1_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
comm-server: TX -> (45): 3a 3c 00 00 00 04 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b 3a 3e
```

```
comm-client: RX <- (45): 3a 3c 00 00 00 04 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (41): 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Server send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)
```

Success Received message on client side is correct (Content {'data.id': 35, 'service.id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on client side): {'data_id': 35, 'service_id': 17, 'status': 4,
↳ 'data': 'msg2_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on client side): result = {'service_id': 17, 'data_id': 35,
↳ 'status': 4, 'data': 'msg2_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)
```

C Test-Coverage

C.1 socket_protocol

The line coverage for socket_protocol was 99.5%

The branch coverage for socket_protocol was 100.0%

C.1.1 socket_protocol.__init__.py

The line coverage for socket_protocol.__init__.py was 99.5%

The branch coverage for socket_protocol.__init__.py was 100.0%

```
1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3 #
4 """
5 socket_protocol (Socket Protocol)
6 =====
7
8 **Author:**
9
10 * Dirk Alders <sudo-dirk@mount-mockery.de>
11
12 **Description:**
13
14     This Module supports point to point communication for client-server issues.
15
16 **Submodules:**
17
```

Unittest for socket_protocol

```
18 * :class:` socket_protocol . data_storage `
19 * :class:` socket_protocol . pure_json_protocol `
20 * :class:` socket_protocol . struct_json_protocol `
21
22 **Unittest:**
23
24     See also the :download:` unittest <socket_protocol/_testresults_/unittest.pdf>`
    documentation .
25
26 **Module Documentation:**
27
28 """
29 __DEPENDENCIES__ = ['stringtools']
30
31 import stringtools
32
33 import binascii
34 import hashlib
35 import json
36 import logging
37 import os
38 import struct
39 import sys
40 import time
41
42
43 try:
44     from config import APP_NAME as ROOT_LOGGER_NAME
45 except ImportError:
46     ROOT_LOGGER_NAME = 'root'
47 logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)
48
49
50 __DESCRIPTION__ = """The Module {\\tt %s} is designed for point to point communication for client
    -server issues .
51 For more Information read the sphinx documentation.""" % __name__.replace('-', '\\-')
52 """The Module Description"""
53 __INTERPRETER__ = (2, 3)
54 """ The Tested Interpreter - Versions """
55
56 SID_AUTH_REQUEST = 0
57 """SID for authentication request"""
58 SID_AUTH_RESPONSE = 1
59 """SID for authentication response"""
60 DID_AUTH_SEED = 0
61 """DID for authentication (seed)"""
62 DID_AUTH_KEY = 1
63 """DID for authentication (key)"""
64 SID_CHANNEL_NAME_REQUEST = 8
65 """SID for channel name exchange request """
66 SID_CHANNEL_NAME_RESPONSE = 9
67 """SID for channel name exchange response"""
68 DID_CHANNEL_NAME = 0
69 """DID for channel name """
70 SID_READ_REQUEST = 10
71 """SID for a read data request"""
72 SID_READ_RESPONSE = 11
73 """SID for read data response"""
74 SID_WRITE_REQUEST = 20
75 """SID for a write data request"""
76 SID_WRITE_RESPONSE = 21
77 """SID for a write data response"""
78 SID_EXECUTE_REQUEST = 30
```

Unittest for socket_protocol

```

79 """SID for a execute request"""
80 SID_EXECUTE_RESPONSE = 31
81 """SID for a execute response"""
82
83 STATUS_OKAY = 0
84 """Status for 'okay'"""
85 STATUS_BUFFERING_UNHANDLED_REQUEST = 1
86 """Status for 'unhandled request'"""
87 STATUS_CALLBACK_ERROR = 2
88 """Status for 'callback errors'"""
89 STATUS_AUTH_REQUIRED = 3
90 """Status for 'authentication is required'"""
91 STATUS_SERVICE_OR_DATA_UNKNOWN = 4
92 """Status for 'service or data unknown'"""
93 STATUS_CHECKSUM_ERROR = 5
94 """Status for 'checksum error'"""
95 STATUS_OPERATION_NOT_PERMITTED = 6
96 """Status for 'operation not permitted'"""
97 STATUS_LOG_LVL = {
98     STATUS_OKAY: logging.INFO,
99     STATUS_BUFFERING_UNHANDLED_REQUEST: logging.WARNING,
100    STATUS_CALLBACK_ERROR: logging.ERROR,
101    STATUS_AUTH_REQUIRED: logging.WARNING,
102    STATUS_SERVICE_OR_DATA_UNKNOWN: logging.ERROR,
103    STATUS_CHECKSUM_ERROR: logging.ERROR,
104    STATUS_OPERATION_NOT_PERMITTED: logging.WARNING,
105 }
106 """Status depending log level for messages"""
107
108 AUTH_STATE_UNTRUSTED_CONNECTION = 0
109 """Authentication Status for an 'Untrusted Connection'"""
110 AUTH_STATE_SEED_REQUESTED = 1
111 """Authentication Status for 'Seed was requested'"""
112 AUTH_STATE_SEED_TRANSFERRED = 2
113 """Authentication Status for 'Seed has been sent'"""
114 AUTH_STATE_KEY_TRANSFERRED = 3
115 """Authentication Status for 'Key has been sent'"""
116 AUTH_STATE_TRUSTED_CONNECTION = 4
117 """Authentication Status for a 'Trusted Connection'"""
118 AUTH_STATE_NAMES = {AUTH_STATE_UNTRUSTED_CONNECTION: 'Untrusted Connection',
119                    AUTH_STATE_SEED_REQUESTED: 'Seed was requested',
120                    AUTH_STATE_SEED_TRANSFERRED: 'Seed has been sent',
121                    AUTH_STATE_KEY_TRANSFERRED: 'Key has been sent',
122                    AUTH_STATE_TRUSTED_CONNECTION: 'Trusted Connection'}
123 """Authentication Status names for previous defined authentication states"""
124
125
126 class RequestSidExistsError(Exception):
127     pass
128
129
130 class ResponseSidExistsError(Exception):
131     pass
132
133
134 class _callback_storage(dict):
135     DEFAULT_CHANNEL_NAME = 'all_others'
136
137     def __init__(self, channel_name, log_prefix):
138         self.init_channel_name(channel_name)
139         self._log_prefix_ = log_prefix
140         dict.__init__(self)
141

```


Unittest for socket_protocol

```

142 def init_channel_name(self, channel_name):
143     if channel_name is None:
144         self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + self.
DEFAULT_CHANNEL_NAME)
145     else:
146         self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' +
channel_name)
147
148 def get(self, service_id, data_id):
149     if dict.get(self, service_id, {}).get(data_id, None) is not None:
150         return self[service_id][data_id]
151     elif dict.get(self, service_id, {}).get(None, None) is not None:
152         return self[service_id][None]
153     elif dict.get(self, None, {}).get(data_id, None) is not None:
154         return self[None][data_id]
155     elif dict.get(self, None, {}).get(None, None) is not None:
156         return self[None][None]
157     else:
158         return (None, None, None)
159
160 def add(self, service_id, data_id, callback, *args, **kwargs):
161     cb_data = self.get(service_id, data_id)
162     if dict.get(self, service_id, {}).get(data_id, None) is not None:
163         if callback is None:
164             self.logger.warning("%s Deleting existing callback %s for service_id (%s) and
data_id (%s)!", self.__log_prefix__(), repr(cb_data[0].__name__), repr(service_id), repr(
data_id))
165             del(self[service_id][data_id])
166             return
167         else:
168             self.logger.warning("%s Overwriting existing callback %s for service_id (%s) and
data_id (%s) to %s!", self.__log_prefix__(), repr(cb_data[0].__name__), repr(service_id),
repr(data_id), repr(callback.__name__))
169     else:
170         self.logger.debug("%s Adding callback %s for SID=%s and DID=%s", self.__log_prefix__
(), repr(callback.__name__), repr(service_id), repr(data_id))
171         if service_id not in self:
172             self[service_id] = {}
173         self[service_id][data_id] = (callback, args, kwargs)
174
175
176 class data_storage(dict):
177     """
178     This is a storage object for socket_protocol messages.
179
180     :param status: The message status.
181     :type status: int
182     :param service_id: The Service-ID.
183     :type service_id: int
184     :param data_id: The Data-ID.
185     :type data_id: int
186     :param data: The transfered data.
187     :type data: any
188     """
189
190     KEY_STATUS = 'status'
191     KEY_SERVICE_ID = 'service_id'
192     KEY_DATA_ID = 'data_id'
193     KEY_DATA = 'data'
194     ALL_KEYS = [KEY_DATA, KEY_DATA_ID, KEY_SERVICE_ID, KEY_STATUS]
195

```

Unittest for socket_protocol

```

196 def __init__(self, *args, **kwargs):
197     dict.__init__(self, *args, **kwargs)
198     for key in self.ALL_KEYS:
199         if key not in self:
200             self[key] = None
201
202 def get_status(self, default=None):
203     """
204     This Method returns the message status.
205
206     :param default: The default value, if no data is available.
207     """
208     return self.get(self.KEY_STATUS, default)
209
210 def get_service_id(self, default=None):
211     """
212     This Method returns the message Service-ID.
213
214     :param default: The default value, if no data is available.
215     """
216     return self.get(self.KEY_SERVICE_ID, default)
217
218 def get_data_id(self, default=None):
219     """
220     This Method returns the message Data-ID.
221
222     :param default: The default value, if no data is available.
223     """
224     return self.get(self.KEY_DATA_ID, default)
225
226 def get_data(self, default=None):
227     """
228     This Method returns the message data.
229
230     :param default: The default value, if no data is available.
231     """
232     return self.get(self.KEY_DATA, default)
233
234
235 class pure_json_protocol(object):
236     """
237     This `class` supports to transfer a message and it's data.
238
239     :param comm_instance: A communication instance.
240     :type comm_instance: instance
241     :param secret: An optional secret (e.g. created by ``binascii.hexlify(os.urandom(24))``).
242     :type secret: str
243     :param auto_auth: An optional parameter to enable (True) automatic authentication,
244     otherwise you need to do it manually, if needed.
245     :type auto_auth: bool
246     :param channel_name: An optional parameter to set a channel name for logging of the
247     communication.
248     :type channel_name: str
249
250     .. hint::
251
252         * The Service-ID is designed to identify the type of the communication (e.g. :const:`
253         READ_REQUEST`, :const:`WRITE_REQUEST`, :const:`READ_RESPONSE`, :const:`WRITE_RESPONSE`, ...)
254         * The Data-ID is designed to identify the requests / responses using the same Service-ID.
255
256     .. note:: The :class:`comm_instance` needs to have at least the following interface:

```

Unittest for socket_protocol

```
255     * A Method :func:`comm_instance.init_channel_name` to set the channel name.
256     * A Constant :const:`comm_instance.IS_CLIENT` to identify that the :class:`comm_instance`
    is a client (True) or a server (False).
257     * A Method :func:`comm_instance.is_connected` to identify if the instance is connected (
    True) or not (False).
258     * A Method :func:`comm_instance.reconnect` to initiate a reconnect.
259     * A Method :func:`comm_instance.register_callback` to register a data available callback.
260     * A Method :func:`comm_instance.register_connect_callback` to register a connect callback
    .
261     * A Method :func:`comm_instance.register_disconnect_callback` to register a disconnect
    callback.
262     * A Method :func:`comm_instance.send` to send data via the :class:`comm_instance`.
263
264 .. note:: The parameter :const:`auto_auth` is only relevant, if a secret is given and the :
    class:`comm_instance` is a client. The authentication is initiated directly after the
    connection is established.
265
266 .. note:: The :const:`channel_name`-exchange will be initiated by the client directly after
    the the connection is established.
267
268     * If a channel_name is given at both communication sides and they are different, the
    client name is taken over and the server will log a warning message.
269
270 **Example:**
271
272 .. literalinclude:: socket_protocol/_examples_/socket_protocol_client.py
273
274 and
275
276 .. literalinclude:: socket_protocol/_examples_/socket_protocol_server.py
277
278 Will result to the following output:
279
280 .. literalinclude:: socket_protocol/_examples_/socket_protocol_client.log
281 """
282 DEFAULT_CHANNEL_NAME = 'all_others'
283
284 def __init__(self, comm_instance, secret=None, auto_auth=False, channel_name=None):
285     self.__comm_inst__ = comm_instance
286     self.__secret__ = secret
287     self.__auto_auth__ = auto_auth
288     #
289     self.__auth_whitelist__ = {}
290     self.__sid_response_dict__ = {}
291     self.__sid_name_dict__ = {}
292     self.__did_name_dict__ = {}
293     #
294     self.__callbacks__ = _callback_storage(channel_name, self.__log_prefix__)
295     self.__init_channel_name__(channel_name)
296     #
297     self.__status_name_dict = {}
298     self.add_status(STATUS_OKAY, 'okay')
299     self.add_status(STATUS_BUFFERING_UNHANDLED_REQUEST, 'no callback for service, data
    buffered')
300     self.add_status(STATUS_CALLBACK_ERROR, 'callback error')
301     self.add_status(STATUS_AUTH_REQUIRED, 'authentication required')
302     self.add_status(STATUS_SERVICE_OR_DATA_UNKNOWN, 'service or data unknown')
303     self.add_status(STATUS_CHECKSUM_ERROR, 'checksum error')
304     self.add_status(STATUS_OPERATION_NOT_PERMITTED, 'operation not permitted')
305     #
306     self.__clean_receive_buffer__()
```

```

308     self.add_service(SID_AUTH_REQUEST, SID_AUTH_RESPONSE, 'authentication request', '
authentication response')
309     self.add_data((SID_AUTH_REQUEST, SID_AUTH_RESPONSE), DID_AUTH_SEED, 'seed')
310     self.add_data(SID_AUTH_REQUEST, DID_AUTH_KEY, 'key')
311     self.add_data(SID_AUTH_RESPONSE, DID_AUTH_KEY, 'key')
312     self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_SEED)
313     self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_SEED)
314     self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_KEY)
315     self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_KEY)
316     self.__callbacks__.add(SID_AUTH_REQUEST, DID_AUTH_SEED, self.
__authenticate_create_seed__)
317     self.__callbacks__.add(SID_AUTH_RESPONSE, DID_AUTH_SEED, self.
__authenticate_create_key__)
318     self.__callbacks__.add(SID_AUTH_REQUEST, DID_AUTH_KEY, self.__authenticate_check_key__)
319     self.__callbacks__.add(SID_AUTH_RESPONSE, DID_AUTH_KEY, self.
__authenticate_process_feedback__)
320     self.__authentication_state_reset__()
321
322     self.add_service(SID_CHANNEL_NAME_REQUEST, SID_CHANNEL_NAME_RESPONSE, 'channel name
request', 'channel name response')
323     self.add_data((SID_CHANNEL_NAME_REQUEST, SID_CHANNEL_NAME_RESPONSE), DID_CHANNEL_NAME, '
name')
324     self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME_REQUEST, DID_CHANNEL_NAME)
325     self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME_RESPONSE, DID_CHANNEL_NAME)
326     self.__callbacks__.add(SID_CHANNEL_NAME_REQUEST, DID_CHANNEL_NAME, self.
__channel_name_request__)
327     self.__callbacks__.add(SID_CHANNEL_NAME_RESPONSE, DID_CHANNEL_NAME, self.
__channel_name_response__)
328
329     self.add_service(SID_READ_REQUEST, SID_READ_RESPONSE, 'read data request', 'read data
response')
330     self.add_service(SID_WRITE_REQUEST, SID_WRITE_RESPONSE, 'write data request', 'write data
response')
331     self.add_service(SID_EXECUTE_REQUEST, SID_EXECUTE_RESPONSE, 'execute request', 'execute
response')
332
333     self.__seed__ = None
334     self.__comm_inst__.register_callback(self.__data_available_callback__)
335     self.__comm_inst__.register_connect_callback(self.__connection_established__)
336     self.__comm_inst__.register_disconnect_callback(self.__authentication_state_reset__)
337     self.logger.info('%s Initialisation finished.', self.__log_prefix__)
338
339     def __analyse_frame__(self, frame):
340         if sys.version_info >= (3, 0):
341             return data_storage(json.loads(frame[:-4].decode('utf-8')))
342         else:
343             return data_storage(json.loads(frame[:-4]))
344
345     def __authenticate_check_key__(self, msg):
346         key = msg.get_data()
347         if key == self.__authenticate_salt_and_hash__(self.__seed__):
348             self.__authentication_state__ = AUTH.STATE_TRUSTED_CONNECTION
349             return STATUS_OKAY, True
350         else:
351             self.__authentication_state__ = AUTH.STATE_UNTRUSTED_CONNECTION
352             return STATUS_OKAY, False
353
354     def __authenticate_create_key__(self, msg):
355         self.__authentication_state__ = AUTH.STATE_KEY_TRANSFERRED
356         seed = msg.get_data()
357         key = self.__authenticate_salt_and_hash__(seed)
358         self.send(SID_AUTH_REQUEST, DID_AUTH_KEY, key)

```

Unittest for socket_protocol

```

359
360 def __authenticate_create_seed__(self, msg):
361     self.__authentication_state__ = AUTH_STATE_SEED_TRANSFERRED
362     if sys.version_info >= (3, 0):
363         self.__seed__ = binascii.hexlify(os.urandom(32)).decode('utf-8')
364     else:
365         self.__seed__ = binascii.hexlify(os.urandom(32))
366     return STATUS_OKAY, self.__seed__
367
368 def __authenticate_process_feedback__(self, msg):
369     feedback = msg.get_data()
370     if feedback:
371         self.__authentication_state__ = AUTH_STATE_TRUSTED_CONNECTION
372         self.logger.info("%s Got positive authentication feedback", self.__log_prefix__())
373     else:
374         self.__authentication_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
375         self.logger.warning("%s Got negative authentication feedback", self.__log_prefix__())
376     return STATUS_OKAY, None
377
378 def __authenticate_salt_and_hash__(self, seed):
379     if sys.version_info >= (3, 0):
380         return hashlib.sha512(bytes(seed, 'utf-8') + self.__secret__).hexdigest()
381     else:
382         return hashlib.sha512(seed.encode('utf-8') + self.__secret__.encode('utf-8')).hexdigest()
383
384 def __authentication_state_reset__(self):
385     self.logger.info("%s Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION", self.__log_prefix__())
386     self.__authentication_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
387
388 def __authentication_required__(self, service_id, data_id):
389     return data_id not in self.__auth_whitelist__.get(service_id, [])
390
391 def __buffer_received_data__(self, msg):
392     if not msg.get_service_id() in self.__msg_buffer__:
393         self.__msg_buffer__[msg.get_service_id()] = {}
394     if not msg.get_data_id() in self.__msg_buffer__[msg.get_service_id()]:
395         self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()] = []
396     self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()].append(msg)
397     self.logger.debug("%s Message data is stored in buffer and is now ready to be retrieved by receive method", self.__log_prefix__())
398
399 def __build_frame__(self, msg):
400     data_frame = json.dumps(self.__mk_msg__(msg.get_status(), msg.get_service_id(), msg.get_data_id(), msg.get_data()))
401     if sys.version_info >= (3, 0):
402         data_frame = bytes(data_frame, 'utf-8')
403     checksum = self.__calc_chksum__(data_frame)
404     return data_frame + checksum
405
406 def __calc_chksum__(self, raw_data):
407     return struct.pack('>I', binascii.crc32(raw_data) & 0xffffffff)
408
409 @property
410 def __channel_name__(self):
411     cn = self.logger.name.split('.')[ -1]
412     if cn != self.DEFAULT_CHANNEL_NAME:
413         return cn
414

```

Unittest for socket_protocol

```

415 def __channel_name_response__(self, msg):
416     data = msg.get_data()
417     if self.__channel_name__ is None and data is not None:
418         self.__init_channel_name__(data)
419         self.logger.info('%s channel name is now %s', self.__log_prefix__(), repr(self.
__channel_name__))
420     return STATUS_OKAY, None
421
422 def __channel_name_request__(self, msg):
423     data = msg.get_data()
424     if data is None:
425         return STATUS_OKAY, self.__channel_name__
426     else:
427         prev_channel_name = self.__channel_name__
428         self.__init_channel_name__(data)
429         if prev_channel_name is not None and prev_channel_name != data:
430             self.logger.warning('%s overwriting user defined channel name from %s to %s',
self.__log_prefix__(), repr(prev_channel_name), repr(data))
431             elif prev_channel_name is None:
432                 self.logger.info('%s channel name is now %s', self.__log_prefix__(), repr(self.
__channel_name__))
433         return STATUS_OKAY, None
434
435 def __check_frame_checksum__(self, frame):
436     return self.__calc_chksum__(frame[:-4]) == frame[-4:]
437
438 def __clean_receive_buffer__(self):
439     self.logger.debug('%s Cleaning up receive-buffer', self.__log_prefix__())
440     self.__msg_buffer__ = {}
441
442 def __connection_established__(self):
443     self.__clean_receive_buffer__()
444     if self.__comm_inst__.IS_CLIENT:
445         self.send(SID_CHANNEL_NAME_REQUEST, 0, self.__channel_name__)
446     if self.__auto_auth__ and self.__comm_inst__.IS_CLIENT and self.__secret__ is not None:
447         self.authenticate()
448
449 def __log_msg__(self, msg, rx_tx_prefix):
450     self.logger.log(
451         self.__status_log_lvl__(msg.get_status()),
452         '%s %s %s, %s, data: "%s"',
453         self.__log_prefix__(),
454         rx_tx_prefix,
455         self.__get_message_name__(msg.get_service_id(), msg.get_data_id()),
456         self.__get_status_name__(msg.get_status()),
457         repr(msg.get_data())
458     )
459
460 def __data_available_callback__(self, comm_inst):
461     frame = comm_inst.receive()
462     msg = self.__analyse_frame__(frame)
463     if not self.__check_frame_checksum__(frame):
464         # Wrong Checksum
465         self.logger.log(self.__status_log_lvl__(STATUS_CHECKSUM_ERROR), '%s Received message
has an invalid checksum. Message will be ignored.', self.__log_prefix__())
466         return # No response needed
467     elif not self.check_authentication_state() and self.__authentication_required__(msg.
get_service_id(), msg.get_data_id()):
468         # Authentication required

```

```

469         self.__log_msg__(msg, 'RX <-')
470         if msg.get_service_id() in self.__sid_response_dict__.keys():
471             self.logger.log(self.__status_log_lvl__(STATUS_AUTH_REQUIRED), "%s
Authentication is required. Just sending negative response.", self.__log_prefix__())
472             status = STATUS_AUTH_REQUIRED
473             data = None
474         else:
475             self.logger.log(self.__status_log_lvl__(STATUS_AUTH_REQUIRED), "%s
Authentication is required. Incoming message will be ignored.", self.__log_prefix__())
476             return # No response needed
477     else:
478         # Valid message
479         self.__log_msg__(msg, 'RX <-')
480         callback, args, kwargs = self.__callbacks__.get(msg.get_service_id(), msg.get_data_id
())
481         if msg.get_service_id() in self.__sid_response_dict__.keys():
482             #
483             # REQUEST RECEIVED
484             #
485             if callback is None:
486                 self.logger.warning("%s Incoming message with no registered callback.
Sending negative response.", self.__log_prefix__())
487                 status = STATUS_BUFFERING_UNHANDLED_REQUEST
488                 data = None
489             else:
490                 self.logger.debug("%s Executing callback %s to process received data", self.
__log_prefix__(), callback.__name__)
491                 try:
492                     status, data = callback(msg, *args, **kwargs)
493                 except Exception as e:
494                     self.logger.error('{lp} Exception raised. Check callback {callback_name}:
"{message}" and it\'s return values for {msg_info}'.format(lp=self.__log_prefix__(),
callback_name=callback.__name__, message=str(e), msg_info=self.__get_message_name__(msg.
get_service_id(), msg.get_data_id())))
495                     status = STATUS_CALLBACK_ERROR
496                     data = None
497             else:
498                 #
499                 # RESPONSE RECEIVED
500                 #
501                 if callback is None:
502                     self.__buffer_received_data__(msg)
503                 else:
504                     self.logger.debug("%s Executing callback %s to process received data", self.
__log_prefix__(), callback.__name__)
505                     try:
506                         callback(msg, *args, **kwargs)
507                     except Exception as e:
508                         self.logger.error('{lp} Exception raised. Check callback {callback_name}:
"{message}" for {msg_info}'.format(lp=self.__log_prefix__(), callback_name=callback.__name__
, message=str(e), msg_info=self.__get_message_name__(msg.get_service_id(), msg.get_data_id()
)))
509                     return # No response needed
510                 self.send(self.__sid_response_dict__[msg.get_service_id()], msg.get_data_id(), data,
status=status)
511
512     def __get_message_name__(self, service_id, data_id):
513         return 'service: %s, data_id: %s' % (
514             self.__sid_name_dict__.get(service_id, repr(service_id)),
515             self.__did_name_dict__.get(service_id, {}).get(data_id, repr(data_id)),
516         )
517

```

Unittest for socket_protocol

```

518 def __get_status_name__(self, status):
519     return 'status: %s' % (self.__status_name_dict.get(status, 'unknown status: %s' % repr(
status)))
520
521 def __init_channel_name__(self, channel_name):
522     self.__comm_inst__.init_channel_name(channel_name)
523     self.__callbacks__.init_channel_name(channel_name)
524     if channel_name is None:
525         self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + self.
DEFAULT_CHANNEL_NAME)
526     else:
527         self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' +
channel_name)
528
529 def __log_prefix__(self):
530     return 'prot-client:' if self.__comm_inst__.IS_CLIENT else 'prot-server:'
531
532 def __mk_msg__(self, status, service_id, data_id, data):
533     return data_storage({data_storage.KEY_DATA.ID: data_id, data_storage.KEY_SERVICE.ID:
service_id, data_storage.KEY_STATUS: status, data_storage.KEY_DATA: data})
534
535 def __status_log_lvl__(self, status):
536     return STATUS_LOG_LVL.get(status, logging.CRITICAL)
537
538 def add_data(self, service_id, data_id, name):
539     """
540     Method to add a name for a specific message.
541
542     :param service_id: The Service-ID of the message. See class definitions starting with ``
SID_``.
543     :type service_id: int or list of ints
544     :param data_id: The Data-ID of the message.
545     :type data_id: int
546     :param name: The Name for the transferred message.
547     :type name: str
548     """
549     try:
550         iter(service_id)
551     except Exception:
552         service_id = (service_id, )
553
554     for sid in service_id:
555         if sid not in self.__did_name_dict__:
556             self.__did_name_dict__[sid] = {}
557             self.__did_name_dict__[sid][data_id] = name
558
559 def add_msg_to_auth_whitelist_(self, service_id, data_id):
560     """
561     Method to add a specific message to the list, where no authentication is required.
562
563     :param service_id: The Service-ID of the message. See class definitions starting with ``
SID_``.
564     :type service_id: int
565     :param data_id: The Data-ID of the message.
566     :type data_id: int
567     """
568     if service_id not in self.__auth_whitelist__:
569         self.__auth_whitelist__[service_id] = []
570     self.__auth_whitelist__[service_id].append(data_id)
571     self.logger.debug('%s Adding Message (%s) to the authentication whitelist', self.
__log_prefix__(), self.__get_message_name__(service_id, data_id))
572

```



```

573 def add_service(self, req_sid, resp_sid, req_name=None, resp_name=None):
574     """
575     Method to add a Service defined by Request- and Response Service-ID.
576
577     :param req_sid: The Request Service-ID.
578     :type req_sid: int
579     :param resp_sid: The Response Service-ID.
580     :type resp_sid: int
581     """
582     if req_sid in self.__sid_response_dict__:
583         self.logger.error('%s Service with Request-SID=%d and Response-SID=%d not added,
584 because request SID is already registered', self.__log_prefix__(), req_sid, resp_sid)
585         raise RequestSidExistsError("Request for this Service is already registered")
586     elif resp_sid in self.__sid_response_dict__.values():
587         self.logger.error('%s Service with Request-SID=%d and Response-SID=%d not added,
588 because response SID is already registered', self.__log_prefix__(), req_sid, resp_sid)
589         raise ResponseSidExistsError("Response for this Service is already registered")
590     else:
591         self.__sid_response_dict__[req_sid] = resp_sid
592         if req_name is not None:
593             self.__sid_name_dict__[req_sid] = req_name
594         if resp_name is not None:
595             self.__sid_name_dict__[resp_sid] = resp_name
596         self.logger.debug('%s Adding Service with Request=%s and Response=%s', self.
597 __log_prefix__(), req_name or repr(req_sid), resp_name or repr(resp_sid))
598
599 def add_status(self, status, name):
600     """
601     Method to add a name for a status.
602
603     :param status: The Status. See class definitions starting with ``STATUS``.
604     :type status: int
605     :param name: The Name for the Status.
606     :type name: str
607     """
608     self.__status_name_dict[status] = name
609
610 def authenticate(self, timeout=2):
611     """
612     This method authenticates the client at the server.
613
614     :param timeout: The timeout for the authentication (requesting seed, sending key and
615 getting authentication_feedback).
616     :type timeout: float
617     :returns: True, if authentication was successfull; False, if not.
618     :rtype: bool
619
620     .. note:: An authentication will only processed, if a secret had been given on
621 initialisation.
622
623     .. note:: Client and Server needs to use the same secret.
624     """
625     if self.__secret__ is not None:
626         self.__authentication_state__ = AUTH.STATE.SEED.REQUESTED
627         self.send(SID.AUTH.REQUEST, DID.AUTH.SEED, None)
628         cnt = 0
629         while cnt < timeout * 10:
630             time.sleep(0.1)
631             if self.__authentication_state__ == AUTH.STATE.TRUSTED.CONNECTION:
632                 return True
633             elif self.__authentication_state__ == AUTH.STATE.UNTRUSTED.CONNECTION:
634                 break
635             cnt += 1
636     return False

```

Unittest for socket_protocol

```

632
633 def check_authentication_state(self):
634     """
635     This Method return the Authentification State as boolean value.
636
637     :return: True, if authentication state is okay, otherwise False
638     :rtype: bool
639     """
640     return self.__secret__ is None or self.__authentication_state__ ==
AUTH.STATE.TRUSTED_CONNECTION
641
642 def connection_established(self):
643     """
644     This Method returns the Connection state including authentication as a boolean value.
645
646     :return: True, if the connection is established (incl. authentication, if a secret has
been given)
647     :rtype: bool
648     """
649     return self.is_connected() and (self.__secret__ is None or self.
check_authentication_state())
650
651 def is_connected(self):
652     """
653     This Methods returns Connection state of the Communication Instance :func:`comm_instance.
is_connected`.
654
655     :return: True if the :class:`comm_instance` is connected, otherwise False..
656     :rtype: bool
657     """
658     return self.__comm_inst__.is_connected()
659
660 def receive(self, service_id, data_id, timeout=1):
661     """
662     This Method returns a message object for a defined message or None, if this message is
not available after the given timeout.
663
664     :param service_id: The Service-ID for the message. See class definitions starting with ``
SID_``.
665     :type service_id: int
666     :param data_id: The Data-ID for the message.
667     :type data_id: int
668     :param timeout: The timeout for receiving.
669     :type timeout: float
670     :returns: The received data storage object or None, if no data was received.
671     :rtype: data_storage
672     """
673     data = None
674     cnt = 0
675     while data is None and cnt < timeout * 10:
676         try:
677             data = self.__msg_buffer__.get(service_id, {}).get(data_id, []).pop(0)
678         except IndexError:
679             data = None
680             cnt += 1
681             time.sleep(0.1)
682     if data is None and cnt >= timeout * 10:
683         self.logger.warning('%s TIMEOUT (%ss): Requested data (service_id: %s; data_id: %s)
not in buffer.', self.__log_prefix__, repr(timeout), repr(service_id), repr(data_id))
684     return data
685
686 def reconnect(self):

```

```

687         """
688         This methods initiates a reconnect by calling :func:`comm_instance.reconnect`.
689         """
690         return self.__comm_inst__.reconnect()
691
692     def register_callback(self, service_id, data_id, callback, *args, **kwargs):
693         """
694         This method registers a callback for the given parameters. Giving ``None`` means, that
695         all Service-IDs or all Data-IDs are used.
696         If a message hitting these parameters has been received, the callback will be executed.
697
698         :param service_id: The Service-ID for the message. See class definitions starting with ``
699         SID_``.
700         :type service_id: int
701         :param data_id: The Data-ID for the message.
702         :type data_id: int
703
704         .. note:: The :func:`callback` is prioritised in the following order:
705
706             * Callbacks with defined Service-ID and Data-ID.
707             * Callbacks with a defined Service-ID and all Data-IDs.
708             * Callbacks with a defined Data-ID and all Service-IDs.
709             * Unspecific Callbacks.
710
711         .. note:: The :func:`callback` is executed with these arguments:
712
713             **Parameters given at the callback call:**
714
715             * The first Arguments is the received message as :class:`data_storage` object.
716             * Further arguments given at registration.
717             * Further keyword arguments given at registration.
718
719             **Return value of the callback:**
720
721             If the Callback is a Request Callback for a registered Service, the return value has
722             to be a tuple or list with
723
724             * :const:`response_status`: The response status (see class definitions starting with
725             :const:`STA_*`).
726             * :const:`response_data`: A JSON iterable object to be used as data for the response.
727
728         .. note:: Only registered services will respond via the callbacks return values with the
729         same data_id.
730         """
731         self.__callbacks__.add(service_id, data_id, callback, *args, **kwargs)
732
733     def send(self, service_id, data_id, data, status=STATUS_OKAY, timeout=2):
734         """
735         This methods sends out a message with the given content.
736
737         :param service_id: The Service-ID for the message. See class definitions starting with ``
738         SERVICE_``.
739         :type service_id: int
740         :param data_id: The Data-ID for the message.
741         :type data_id: int
742         :param data: The data to be transfered. The data needs to be json compatible.
743         :type data: str
744         :param status: The Status for the message. All requests should have ``STATUS_OKAY``.
745         :type status: int
746         :param timeout: The timeout for sending data (e.g. time to establish new connection).
747         :type timeout: float

```

Unittest for socket_protocol

```

742         :return: True if data had been sent, otherwise False.
743         :rtype: bool
744         """
745         if (self.check_authentication_state() or not self.__authentication_required__(
            service_id, data_id)) or (service_id in self.__sid_response_dict__.values() and status ==
            STATUS_AUTH_REQUIRED and data is None):
746             msg = data_storage(service_id=service_id, data_id=data_id, data=data, status=status)
747             self.__log_msg__(msg, 'TX ->')
748             return self.__comm_inst__.send(self.__build_frame__(msg), timeout=timeout)
749         else:
750             # Authentication required
751             self.logger.warning("%s Authentication is required. TX-Message %s, %s, data: %s
            will be ignored.", self.__log_prefix__(), self.__get_message_name__(service_id, data_id),
            self.__get_status_name__(status), repr(data))
752             return False
753
754
755 class struct_json_protocol(pure_json_protocol):
756     """
757     This Class has the same functionality like :class:`pure_json_protocol`. The message length is
758     less than for :class:`pure_json_protocol`, but the functionality and compatibility is
759     reduced.
760     See also parent :py:class:`pure_json_protocol`.
761
762     .. note::
763         This class is depreceated and here for compatibility reasons (to support old clients or
764         servers). Usage of :class:`pure_json_protocol` is recommended.
765     """
766     def __init__(self, *args, **kwargs):
767         pure_json_protocol.__init__(self, *args, **kwargs)
768
769     def __analyse_frame__(self, frame):
770         status, service_id, data_id = struct.unpack('>III', frame[0:12])
771         if sys.version_info >= (3, 0):
772             data = json.loads(frame[12:-1].decode('utf-8'))
773         else:
774             data = json.loads(frame[12:-1])
775         return self.__mk_msg__(status, service_id, data_id, data)
776
777     def __build_frame__(self, msg):
778         frame = struct.pack('>III', msg.get_status(), msg.get_service_id(), msg.get_data_id())
779         if sys.version_info >= (3, 0):
780             frame += bytes(json.dumps(msg.get_data()), 'utf-8')
781             frame += self.__calc_chksum__(frame)
782         else:
783             frame += json.dumps(msg.get_data())
784             frame += self.__calc_chksum__(frame)
785         return frame
786
787     def __calc_chksum__(self, raw_data):
788         chksum = 0
789         for b in raw_data:
790             if sys.version_info >= (3, 0):
791                 chksum ^= b
792             else:
793                 chksum ^= ord(b)
794         if sys.version_info >= (3, 0):
795             return bytes([chksum])
796         else:
797             return chr(chksum)
798
799     def __check_frame_checksum__(self, frame):
800         return self.__calc_chksum__(frame[:-1]) == frame[-1:]

```