Unittest for `socket_protocol`

August 15, 2025

Contents

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	python3
Version	2e0ebc3d214d2d8e9e4eaca3d7d1b3bf
Dependencies	

1.2 Unittest Information

Unittest Information	
Version	e2f899a8597d10f58dc24606f3b9dcfe
Testruns with	python 3.13.5 (final)

1.3 Test System Information

System Information	
Architecture	64bit
Distribution	Debian GNU/Linux 13 trixie
Hostname	ahorn
Kernel	6.12.38+deb13-amd64 (#1 SMP PREEMPT_DYNAMIC Debian 6.12.38-1 (2025-07-16))
Machine	x86_64
Path	/home/dirk/work/unittest_collection/socket_protocol
System	Linux
Username	dirk

2 Statistic

2.1 Test-Statistic for testrun with python 3.13.5 (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	19.298s

2.2 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
socket_protocol	99.5%	98.9%
socket_protocol.__init__.py	99.5%	

3 Tested Requirements

3.1 Message Object

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, Authentification 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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:41:51,106
Finished-Time:	2025-08-15 21:41:51,107
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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:41:51,107
Finished-Time:	2025-08-15 21:41:51,108
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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:41:51,108
Finished-Time:	2025-08-15 21:41:51,109
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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:41:51,109
Finished-Time:	2025-08-15 21:41:51,110
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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:41:51,110
Finished-Time:	2025-08-15 21:41:51,868
Time-Consumption	0.759s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).
Info	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).

3.2.2 A checksumm shall ensure the correct transmition

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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:41:51,869
Finished-Time:	2025-08-15 21:41:52,836
Time-Consumption	0.967s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transfering 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	Transfering 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 authentification between server and client shall be possible including status feedback methods

Description

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

Reason for the implementation

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

Fitcriterion

Check authentification method feedback (client) and authentification 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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:41:52,837
Finished-Time:	2025-08-15 21:41:53,830
Time-Consumption	0.994s

Testsummary:

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

Success	Authentification state of client is correct (Content True and Type is <class 'bool'>).
Info	Corrupting the authentification mechanism
Info	Performing Authentification
Success	Return Value of authentification method is correct (Content False and Type is <class 'bool'>).
Success	Authentification state of server is correct (Content False and Type is <class 'bool'>).
Success	Authentification state of client is correct (Content False and Type is <class 'bool'>).

3.2.4 An automatic authentification shall available

Description

An authentification is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentification.

Fitcriterion

Check authentification 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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:41:53,832
Finished-Time:	2025-08-15 21:41:56,548
Time-Consumption	2.717s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Identical secrets set and automatic authentification
Success	Authentification state of server is correct (Content False and Type is <class 'bool'>).
Success	Authentification state of client is correct (Content False and Type is <class 'bool'>).
Info	Connecting Server and Client
Success	Authentification state of server is correct (Content True and Type is <class 'bool'>).
Success	Authentification 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 authentification 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 Authentification 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 authentification has been performed.

Testresult

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

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:41:56,549
Finished-Time:	2025-08-15 21:41:58,526
Time-Consumption	1.977s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Setting a Server secret and no Client secret
Info	Transfering 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	Transfering 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	Transfering 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	Transfering 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 Authentification
Info	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).
Info	Transfering 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_transfered'} 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 authentification was successfull performed.

Reason for the implementation

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

Fitcriterion

Transmition 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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:41:58,526
Finished-Time:	2025-08-15 21:42:01,007
Time-Consumption	2.481s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Identical secrets set
Info	Transfering 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	Transfering 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	Transfering 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	Transfering 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	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).
Info	Transfering 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	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).
Info	Transfering 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_transfered'} 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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:01,008
Finished-Time:	2025-08-15 21:42:02,747
Time-Consumption	1.739s

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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:02,748
Finished-Time:	2025-08-15 21:42:03,609
Time-Consumption	0.861s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transfering 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	Transfering 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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:03,609
Finished-Time:	2025-08-15 21:42:03,966
Time-Consumption	0.356s

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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:03,966
Finished-Time:	2025-08-15 21:42:04,932
Time-Consumption	0.966s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transfering 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	Transfering 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	Transfering 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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:04,934
Finished-Time:	2025-08-15 21:42:05,496
Time-Consumption	0.562s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transfering 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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:05,496
Finished-Time:	2025-08-15 21:42:06,052
Time-Consumption	0.555s

Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transfering 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 incomming messages

Testresult

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

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:06,053
Finished-Time:	2025-08-15 21:42:06,618
Time-Consumption	0.566s

Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transfering 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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:06,619
Finished-Time:	2025-08-15 21:42:07,786
Time-Consumption	1.167s

Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering all kind of Callbacks
Info	Transfering 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	Transfering 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	Transfering 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 Serice-IDs
Info	Transfering 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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:07,786
Finished-Time:	2025-08-15 21:42:08,589
Time-Consumption	0.803s

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 authentification
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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:08,590
Finished-Time:	2025-08-15 21:42:08,942
Time-Consumption	0.352s

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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:08,942
Finished-Time:	2025-08-15 21:42:09,650
Time-Consumption	0.708s

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 Depreceated 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 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (331)
Start-Time:	2025-08-15 21:42:09,651
Finished-Time:	2025-08-15 21:42:10,415
Time-Consumption	0.765s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).
Info	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).

A Trace for testrun with python 3.13.5 (final)

A.1 Tests with status Info (22)

A.1.1 REQ-0001

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'>)
```

A.1.2 REQ-0002

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'>)
```

A.1.3 REQ-0003

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'>)
```

A.1.4 REQ-0004

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'>)
```

A.1.5 REQ-0005

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
```

```
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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
```

```
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
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering 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 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
```

```
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
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
```

```
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

A.1.6 REQ-0006

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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
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
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 Transfering 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 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
```

```
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
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
```

```
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'>)
```

A.1.7 REQ-0007

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 Authentification

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

```
Result (Return Value of authentification method): False (<class 'bool'>)
Expectation (Return Value of authentification method): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of server): True (<class 'bool'>)
Expectation (Authentification state of server): result = True (<class 'bool'>)
```

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

```
Result (Authentification state of client): True (<class 'bool'>)
Expectation (Authentification state of client): result = True (<class 'bool'>)
```

Info Different secrets set

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

```
Result (Authentification state of server): False (<class 'bool'>)
Expectation (Authentification state of server): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of client): False (<class 'bool'>)
Expectation (Authentification state of client): result = False (<class 'bool'>)
```

Info Performing Authentification

```
prot-client: TX -> service: authentification 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
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
prot-server: RX <- service: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'be1ce6e6a7798e984e4282fb73f762002a47e898323f3197afbe2bd75824312e'"
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 62 65 31 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 62 65 31 63
comm-server: TX -> (64): 65 36 65 36 61 37 37 39 38 65 39 38 34 65 34 32 38 32 66 62 37 33 66
↪  37 36 32 30 30 32 61 34 37 65 38 39 38 33 32 33 66 33 31 39 37 61 66 62 65 32 62 64 37 35
↪  38 32 34 33 31 32 65 22 7d 29 6d
comm-client: RX <- (64): 65 36 65 36 61 37 37 39 38 65 39 38 34 65 34 32 38 32 66 62 37 33 66
↪  37 36 32 30 30 32 61 34 37 65 38 39 38 33 32 33 66 33 31 39 37 61 66 62 65 32 62 64 37 35
↪  38 32 34 33 31 32 65 22 7d 29 6d
comm-server: TX -> (4): 16 e4 3a 3e
comm-client: RX <- (4): 16 e4 3a 3e
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'be1ce6e6a7798e984e4282fb73f762002a47e898323f3197afbe2bd75824312e'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'d11fc0b6a06e1536b25ec75c43999f4c2a5a4d3372ba4487cefd5448c104bc8155b2c69a9c131b20bfd7b5b
↪  5aefa3899d0e43344738cdab4e64e3b55ba172626'"
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 64 31 31 66
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 64 31 31 66
comm-client: TX -> (64): 63 30 62 36 61 30 36 65 31 35 33 36 62 32 35 65 63 37 35 63 34 33 39
↪  39 39 66 34 63 32 61 35 61 34 64 33 33 37 32 62 61 34 34 38 37 63 65 66 64 35 34 34 38 63
↪  31 30 34 62 63 38 31 35 35 62 32
comm-server: RX <- (64): 63 30 62 36 61 30 36 65 31 35 33 36 62 32 35 65 63 37 35 63 34 33 39
↪  39 39 66 34 63 32 61 35 61 34 64 33 33 37 32 62 61 34 34 38 37 63 65 66 64 35 34 34 38 63
↪  31 30 34 62 63 38 31 35 35 62 32
```

```
comm-client: TX -> (64): 63 36 39 61 39 63 31 33 31 62 32 30 62 66 64 37 62 35 62 35 61 65 66
↪  61 33 38 39 39 64 30 65 34 33 33 34 34 37 33 38 63 64 61 62 34 65 36 34 65 33 62 35 35 62
↪  61 31 37 32 36 32 36 22 7d 60 ed
comm-server: RX <- (64): 63 36 39 61 39 63 31 33 31 62 32 30 62 66 64 37 62 35 62 35 61 65 66
↪  61 33 38 39 39 64 30 65 34 33 33 34 34 37 33 38 63 64 61 62 34 65 36 34 65 33 62 35 35 62
↪  61 31 37 32 36 32 36 22 7d 60 ed
comm-client: TX -> (4): ad 67 3a 3e
comm-server: RX <- (4): ad 67 3a 3e
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'d11fc0b6a06e1536b25ec75c43999f4c2a5a4d3372ba4487cefd5448c104bc8155b2c69a9c131b20bfd7b5b
↪  5aefa3899d0e43344738cdab4e64e3b55ba172626'"
prot-server: Executing callback __authentificate_check_key__ to process received data
prot-server: TX -> service: authentification 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
comm-server: TX -> (7): 7d ea 0a 5c b4 3a 3e
comm-client: RX <- (7): 7d ea 0a 5c b4 3a 3e
prot-client: RX <- service: authentification response, data_id: key, status: okay, data:
↪  "False"
prot-client: Executing callback __authentificate_process_feedback__ to process received data
prot-client: Got negative authentification feedback
```

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

```
Result (Return Value of authentification method): False (<class 'bool'>)
Expectation (Return Value of authentification method): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of server): False (<class 'bool'>)
Expectation (Authentification state of server): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of client): False (<class 'bool'>)
Expectation (Authentification state of client): result = False (<class 'bool'>)
```

Info Identical secrets set

Info Performing Authentification

```
prot-client: TX -> service: authentification 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
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
prot-server: RX <- service: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'2fe61a51bfa7a30b0521558aaab9db58a190d59ec092a424e71d17eff8554721'"
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 32 66 65 36
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 32 66 65 36
comm-server: TX -> (64): 31 61 35 31 62 66 61 37 61 33 30 62 30 35 32 31 35 35 38 61 61 61 62
↪  39 64 62 35 38 61 31 39 30 64 35 39 65 63 30 39 32 61 34 32 34 65 37 31 64 31 37 65 66 66
↪  38 35 35 34 37 32 31 22 7d 48 1d
comm-client: RX <- (64): 31 61 35 31 62 66 61 37 61 33 30 62 30 35 32 31 35 35 38 61 61 61 62
↪  39 64 62 35 38 61 31 39 30 64 35 39 65 63 30 39 32 61 34 32 34 65 37 31 64 31 37 65 66 66
↪  38 35 35 34 37 32 31 22 7d 48 1d
comm-server: TX -> (4): 99 16 3a 3e
comm-client: RX <- (4): 99 16 3a 3e
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'2fe61a51bfa7a30b0521558aaab9db58a190d59ec092a424e71d17eff8554721'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'d68aed550e0fc34696097bd084368df899f8709ac2eeda88098b6494148fe87dcad55de4fcfa0697114222e
↪  d01897f4697df6215de97f8ec0810762209754247'"
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 64 36 38 61
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 64 36 38 61
comm-client: TX -> (64): 65 64 35 35 30 65 30 66 63 33 34 36 39 36 30 39 37 62 64 30 38 34 33
↪  36 38 64 66 38 39 39 66 38 37 30 39 61 63 32 65 65 64 61 38 38 30 39 38 62 36 34 39 34 31
↪  34 38 66 65 38 37 64 63 61 64 35
comm-server: RX <- (64): 65 64 35 35 30 65 30 66 63 33 34 36 39 36 30 39 37 62 64 30 38 34 33
↪  36 38 64 66 38 39 39 66 38 37 30 39 61 63 32 65 65 64 61 38 38 30 39 38 62 36 34 39 34 31
↪  34 38 66 65 38 37 64 63 61 64 35
```

```
comm-client: TX -> (64): 35 64 65 34 66 63 66 61 30 36 39 37 31 31 34 32 32 32 65 64 30 31 38
↪  39 37 66 34 36 39 37 64 66 36 32 31 35 64 65 39 37 66 38 65 63 30 38 31 30 37 36 32 32 30
↪  39 37 35 34 32 34 37 22 7d 06 2b
```

```
comm-server: RX <- (64): 35 64 65 34 66 63 66 61 30 36 39 37 31 31 34 32 32 32 65 64 30 31 38
↪  39 37 66 34 36 39 37 64 66 36 32 31 35 64 65 39 37 66 38 65 63 30 38 31 30 37 36 32 32 30
↪  39 37 35 34 32 34 37 22 7d 06 2b
```

```
comm-client: TX -> (4): 92 c1 3a 3e
```

```
comm-server: RX <- (4): 92 c1 3a 3e
```

```
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'d68aed550e0fc34696097bd084368df899f8709ac2eeda88098b6494148fe87dcad55de4fcfa0697114222e
↪  d01897f4697df6215de97f8ec0810762209754247'"
```

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

```
prot-server: TX -> service: authentification 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
```

```
comm-server: TX -> (6): 94 fe 74 32 3a 3e
```

```
comm-client: RX <- (6): 94 fe 74 32 3a 3e
```

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

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

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

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

```
Result (Return Value of authentification method): True (<class 'bool'>)
```

```
Expectation (Return Value of authentification method): result = True (<class 'bool'>)
```

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

```
Result (Authentification state of server): True (<class 'bool'>)
```

```
Expectation (Authentification state of server): result = True (<class 'bool'>)
```

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

```
Result (Authentification state of client): True (<class 'bool'>)
```

```
Expectation (Authentification state of client): result = True (<class 'bool'>)
```

Info Corrupting the authentification mechanism

```
prot-server: Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
```

```
prot-client: Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
```

Info Performing Authentification

```
prot-client: TX -> service: authentification 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
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
prot-server: RX <- service: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
```

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

```
Result (Return Value of authentification method): False (<class 'bool'>)
Expectation (Return Value of authentification method): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of server): False (<class 'bool'>)
Expectation (Authentification state of server): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of client): False (<class 'bool'>)
Expectation (Authentification state of client): result = False (<class 'bool'>)
```

A.1.8 REQ-0014

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=authentification request and
↪  Response=authentification response
```

```
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 authentification

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

```
Result (Authentification state of server): False (<class 'bool'>)
Expectation (Authentification state of server): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of client): False (<class 'bool'>)
Expectation (Authentification state of client): result = False (<class 'bool'>)
```

Info Connecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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: authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
prot-server: RX <- service: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'9503fc309ebe8f46fce8ca28e02323f4ac86c4cfbb586cbe591d2f1c84218330'"
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 39 35 30 33
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 35 30 33
comm-server: TX -> (64): 66 63 33 30 39 65 62 65 38 66 34 36 66 63 65 38 63 61 32 38 65 30 32
↪  33 32 33 66 34 61 63 38 36 63 34 63 66 62 62 35 38 36 63 62 65 35 39 31 64 32 66 31 63 38
↪  34 32 31 38 33 33 30 22 7d 02 39
comm-client: RX <- (64): 66 63 33 30 39 65 62 65 38 66 34 36 66 63 65 38 63 61 32 38 65 30 32
↪  33 32 33 66 34 61 63 38 36 63 34 63 66 62 62 35 38 36 63 62 65 35 39 31 64 32 66 31 63 38
↪  34 32 31 38 33 33 30 22 7d 02 39
comm-server: TX -> (4): cd 2b 3a 3e
comm-client: RX <- (4): cd 2b 3a 3e
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'9503fc309ebe8f46fce8ca28e02323f4ac86c4cfbb586cbe591d2f1c84218330'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'7581eb5b21ea16f4e15384d9b4130c9237cbf5c1f648849add4f9c97a365ae23b0715a55ba9b64c86ddd35f
↪  e1975b11c4fb24c9f668d67da453814bbf2496bc3'"
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 35 38 31
```

```
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 35 38 31
```

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

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

```
comm-client: TX -> (64): 35 61 35 35 62 61 39 62 36 34 63 38 36 64 64 64 33 35 66 65 31 39 37
↪  35 62 31 31 63 34 66 62 32 34 63 39 66 36 36 38 64 36 37 64 61 34 35 33 38 31 34 62 62 66
↪  32 34 39 36 62 63 33 22 7d 6b b6
```

```
comm-server: RX <- (64): 35 61 35 35 62 61 39 62 36 34 63 38 36 64 64 64 33 35 66 65 31 39 37
↪  35 62 31 31 63 34 66 62 32 34 63 39 66 36 36 38 64 36 37 64 61 34 35 33 38 31 34 62 62 66
↪  32 34 39 36 62 63 33 22 7d 6b b6
```

```
comm-client: TX -> (4): 22 9d 3a 3e
```

```
comm-server: RX <- (4): 22 9d 3a 3e
```

```
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'7581eb5b21ea16f4e15384d9b4130c9237cbf5c1f648849add4f9c97a365ae23b0715a55ba9b64c86ddd35f
↪  e1975b11c4fb24c9f668d67da453814bbf2496bc3'"
```

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

```
prot-server: TX -> service: authentification 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
```

```
comm-server: TX -> (6): 94 fe 74 32 3a 3e
```

```
comm-client: RX <- (6): 94 fe 74 32 3a 3e
```

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

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

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

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

```
Result (Authentification state of server): True (<class 'bool'>)
```

```
Expectation (Authentification state of server): result = True (<class 'bool'>)
```

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

```
Result (Authentification state of client): True (<class 'bool'>)
```

```
Expectation (Authentification state of client): result = True (<class 'bool'>)
```

A.1.9 REQ-0008

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 Transfering a message client → server

```
prot-client: TX -> service: execute request, data_id: 36, status: okay, data:
↪  "'msg3_data_to_be_transfered'"
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
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
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
prot-server: RX <- service: execute request, data_id: 36, status: okay, data:
↪  "'msg3_data_to_be_transfered'"
prot-server: Authentification is required. Just sending negative response.
prot-server: TX -> service: execute response, data_id: 36, status: authentification 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
comm-server: TX -> (8): 6c 7d 5d 78 af a4 3a 3e
comm-client: RX <- (8): 6c 7d 5d 78 af a4 3a 3e
prot-client: RX <- service: execute response, data_id: 36, status: authentification 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 Transfering 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 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
```

```
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
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
```

```
prot-client: Authentification 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 Transfering a message client → server

```
prot-client: Authentification 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 <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 Transfering a message server → client

```
prot-server: Authentification 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 <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 Authentification

```
prot-client: TX -> service: authentification 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
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
prot-server: RX <- service: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'cf7de519476e5a0a037bb6820cf86a79af9d49ac04b7aae0fdcb1a175fbb1027'"
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 66 37 64
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 66 37 64
comm-server: TX -> (64): 65 35 31 39 34 37 36 65 35 61 30 61 30 33 37 62 62 36 38 32 30 63 66
↪  38 36 61 37 39 61 66 39 64 34 39 61 63 30 34 62 37 61 61 65 30 66 64 63 62 31 61 31 37 35
↪  66 62 62 31 30 32 37 22 7d c6 4a
comm-client: RX <- (64): 65 35 31 39 34 37 36 65 35 61 30 61 30 33 37 62 62 36 38 32 30 63 66
↪  38 36 61 37 39 61 66 39 64 34 39 61 63 30 34 62 37 61 61 65 30 66 64 63 62 31 61 31 37 35
↪  66 62 62 31 30 32 37 22 7d c6 4a
comm-server: TX -> (4): 1e e9 3a 3e
comm-client: RX <- (4): 1e e9 3a 3e
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'cf7de519476e5a0a037bb6820cf86a79af9d49ac04b7aae0fdcb1a175fbb1027'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'f022b79a9666c79993f00d7798d1762b9ab18a29afec5154b60f7810cb19779ab9fa9c85c68fdd5441a1c95
↪  c639757b19166e2564533db1f7404adb30df07d9e'"
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 66 30 32 32
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 66 30 32 32
comm-client: TX -> (64): 62 37 39 61 39 36 36 36 63 37 39 39 39 33 66 30 30 64 37 37 39 38 64
↪  31 37 36 32 62 39 61 62 31 38 61 32 39 61 66 65 63 35 31 35 34 62 36 30 66 37 38 31 30 63
↪  62 31 39 37 37 39 61 62 39 66 61
comm-server: RX <- (64): 62 37 39 61 39 36 36 36 63 37 39 39 39 33 66 30 30 64 37 37 39 38 64
↪  31 37 36 32 62 39 61 62 31 38 61 32 39 61 66 65 63 35 31 35 34 62 36 30 66 37 38 31 30 63
↪  62 31 39 37 37 39 61 62 39 66 61
comm-client: TX -> (64): 39 63 38 35 63 36 38 66 64 64 35 34 34 31 61 31 63 39 35 63 36 33 39
↪  37 35 37 62 31 39 31 36 36 65 32 35 36 34 35 33 33 64 62 31 66 37 34 30 34 61 64 62 33 30
↪  64 66 30 37 64 39 65 22 7d f7 aa
```

```
comm-server: RX <- (64): 39 63 38 35 63 36 38 66 64 64 35 34 34 31 61 31 63 39 35 63 36 33 39
↪  37 35 37 62 31 39 31 36 36 65 32 35 36 34 35 33 33 64 62 31 66 37 34 30 34 61 64 62 33 30
↪  64 66 30 37 64 39 65 22 7d f7 aa
comm-client: TX -> (4): 9f 01 3a 3e
comm-server: RX <- (4): 9f 01 3a 3e
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'f022b79a9666c79993f00d7798d1762b9ab18a29afec5154b60f7810cb19779ab9fa9c85c68fdd5441a1c95
↪  c639757b19166e2564533db1f7404adb30df07d9e'"
prot-server: Executing callback __authentificate_check_key__ to process received data
prot-server: TX -> service: authentification 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
comm-server: TX -> (6): 94 fe 74 32 3a 3e
comm-client: RX <- (6): 94 fe 74 32 3a 3e
prot-client: RX <- service: authentification response, data_id: key, status: okay, data:
↪  "True"
prot-client: Executing callback __authentificate_process_feedback__ to process received data
prot-client: Got positive authentification feedback
```

Info Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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
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
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering 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 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
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
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

A.1.10 REQ-0009

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
```

```
comm-server: TX -> (6): 30 59 be 2f 3a 3e
```

```
comm-client: RX <- (6): 30 59 be 2f 3a 3e
```

```
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 Transfering a message client → server

```
prot-client: Authentification 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 <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 Transfering a message server → client

```
prot-server: Authentification 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 <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 authentification whitelist
```

Info Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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
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
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
prot-server: Authentification 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 Transfering a message server → client

```
prot-server: Authentification 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 <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 authentification whitelist
```

Info Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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
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
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering a message server → client

```
prot-server: Authentification 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 <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 authentification whitelist
prot-server: Adding Message (service: 17, data_id: 35) to the authentification whitelist
```

Info Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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
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
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering 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 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
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
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

A.1.11 REQ-0010

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
```

```
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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
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
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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
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
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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
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
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
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
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'>)
```

A.1.12 REQ-0011

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
```

```
comm-server: TX -> (6): 30 59 be 2f 3a 3e
```

```
comm-client: RX <- (6): 30 59 be 2f 3a 3e
```

```
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 Transfering a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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
```

```
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
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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 Transfering a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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
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
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
prot-server: Incomming 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
comm-server: TX -> (8): 6c 7d bd 30 46 9b 3a 3e
comm-client: RX <- (8): 6c 7d bd 30 46 9b 3a 3e
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'>)
```

A.1.13 REQ-0012

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 REQ-0013

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
```

```
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 Transfering 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
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
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 Transfering 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
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (7): 7d a1 a2 87 f3 3a 3e
comm-client: RX <- (7): 7d a1 a2 87 f3 3a 3e
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 Transfering 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
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (7): 7d 88 6a 33 01 3a 3e
comm-client: RX <- (7): 7d 88 6a 33 01 3a 3e
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'>)
```

A.1.15 REQ-0015

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 Transfering 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
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
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'>)
```

A.1.16 REQ-0016

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 Transfering 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
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
```

```
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
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'>)
```

A.1.17 REQ-0017

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=authentification request and
↪ Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪ authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
```

```
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 Transfering 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
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
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'>)
```

A.1.18 REQ-0018

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
```

```
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 Transfering 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
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
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 Transfering 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
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): 62 51 ca 3a 3e
comm-client: RX <- (5): 62 51 ca 3a 3e
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 Transfering 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
```

```
comm-client: TX -> (5): 5b f5 78 3a 3e
```

```
comm-server: RX <- (5): 5b f5 78 3a 3e
```

```
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
```

```
comm-server: TX -> (5): 79 60 8b 3a 3e
```

```
comm-client: RX <- (5): 79 60 8b 3a 3e
```

```
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 Transfering 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
```

```
comm-client: TX -> (5): 5b f5 78 3a 3e
```

```
comm-server: RX <- (5): 5b f5 78 3a 3e
```

```
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
```

```
comm-server: TX -> (5): 96 78 fe 3a 3e
```

```
comm-client: RX <- (5): 96 78 fe 3a 3e
```

```
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'>)
```

A.1.19 REQ-0020

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 authentification

```
prot-client: TX -> service: authentification 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
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
prot-server: RX <- service: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'f4e143f54a94b1589bbd7289d1706996fd8a474705a545686dcc650807ac50b0'"
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 66 34 65 31
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 66 34 65 31
comm-server: TX -> (64): 34 33 66 35 34 61 39 34 62 31 35 38 39 62 62 64 37 32 38 39 64 31 37
↪  30 36 39 39 36 66 64 38 61 34 37 34 37 30 35 61 35 34 35 36 38 36 64 63 63 36 35 30 38 30
↪  37 61 63 35 30 62 30 22 7d a9 a2
comm-client: RX <- (64): 34 33 66 35 34 61 39 34 62 31 35 38 39 62 62 64 37 32 38 39 64 31 37
↪  30 36 39 39 36 66 64 38 61 34 37 34 37 30 35 61 35 34 35 36 38 36 64 63 63 36 35 30 38 30
↪  37 61 63 35 30 62 30 22 7d a9 a2
comm-server: TX -> (4): 38 06 3a 3e
```

```
comm-client: RX <- (4): 38 06 3a 3e
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'f4e143f54a94b1589bbd7289d1706996fd8a474705a545686dcc650807ac50b0'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'3252f8546f647f33ab0dfb8dcbb09d1657c9f81c2a28695f08078e83fa5b52cee554a6359db7b4aa744ff53
↪  1acb7d1615c550809efd1aedfe614e0ce09a9f5f1'"
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 33 32 35 32
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 33 32 35 32
comm-client: TX -> (64): 66 38 35 34 36 66 36 34 37 66 33 33 61 62 30 64 66 62 38 64 63 62 62
↪  30 39 64 31 36 35 37 63 39 66 38 31 63 32 61 32 38 36 39 35 66 30 38 30 37 38 65 38 33 66
↪  61 35 62 35 32 63 65 65 35 35 34
comm-server: RX <- (64): 66 38 35 34 36 66 36 34 37 66 33 33 61 62 30 64 66 62 38 64 63 62 62
↪  30 39 64 31 36 35 37 63 39 66 38 31 63 32 61 32 38 36 39 35 66 30 38 30 37 38 65 38 33 66
↪  61 35 62 35 32 63 65 65 35 35 34
comm-client: TX -> (64): 61 36 33 35 39 64 62 37 62 34 61 61 37 34 34 66 66 35 33 31 61 63 62
↪  37 64 31 36 31 35 63 35 35 30 38 30 39 65 66 64 31 61 65 64 66 65 36 31 34 65 30 63 65 30
↪  39 61 39 66 35 66 31 22 7d 9f ca
comm-server: RX <- (64): 61 36 33 35 39 64 62 37 62 34 61 61 37 34 34 66 66 35 33 31 61 63 62
↪  37 64 31 36 31 35 63 35 35 30 38 30 39 65 66 64 31 61 65 64 66 65 36 31 34 65 30 63 65 30
↪  39 61 39 66 35 66 31 22 7d 9f ca
comm-client: TX -> (4): 0c c0 3a 3e
comm-server: RX <- (4): 0c c0 3a 3e
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'3252f8546f647f33ab0dfb8dcbb09d1657c9f81c2a28695f08078e83fa5b52cee554a6359db7b4aa744ff53
↪  1acb7d1615c550809efd1aedfe614e0ce09a9f5f1'"
prot-server: Executing callback __authentificate_check_key__ to process received data
prot-server: TX -> service: authentification 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
comm-server: TX -> (6): 94 fe 74 32 3a 3e
comm-client: RX <- (6): 94 fe 74 32 3a 3e
prot-client: RX <- service: authentification response, data_id: key, status: okay, data:
↪  "True"
prot-client: Executing callback __authentificate_process_feedback__ to process received data
```

```
prot-client: Got positive authentification 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'>)
```

A.1.20 REQ-0021

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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'>)
```

A.1.21 REQ-0022

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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
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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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'>)
```

A.1.22 REQ-0023

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 -> (21): 3a 3c 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13 3a 3e
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
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
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
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 Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
comm-client: TX -> (45): 3a 3c 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 3a 3e
comm-server: RX <- (45): 3a 3c 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 3a 3e
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering 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 -> (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
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

B Test-Coverage

B.1 socket_protocol

The line coverage for socket_protocol was 99.5%
The branch coverage for socket_protocol was 98.9%

B.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 98.9%

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
"""
socket_protocol (Socket Protocol)
=================================

**Author:**

* Dirk Alders <sudo-dirk@mount-mockery.de>

**Description:**

    This Module supports point to point communication for client-server issues.

**Submodules:**

* :class:`socket_protocol.data_storage`
* :class:`socket_protocol.pure_json_protocol`
* :class:`socket_protocol.struct_json_protocol`

**Unittest:**

        See also the :download:`unittest <socket_protocol/_testresults_/unittest.pdf>`
    documentation.

**Module Documentation:**

"""
__DEPENDENCIES__ = []

import binascii
import hashlib
import json
import logging
import os
import struct
import time


try:
    from config import APP_NAME as ROOT_LOGGER_NAME
except ImportError:
    ROOT_LOGGER_NAME = 'root'
logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)


__DESCRIPTION__ = """The Module {\\tt %s} is designed for point to point communication for client
    -server issues.
For more Information read the sphinx documentation.""" % __name__.replace('_', '\\_')
"""The Module Description"""
__INTERPRETER__ = (3, )
"""The Tested Interpreter-Versions"""

```

```
SID_AUTH_REQUEST = 0
"""SID for authentification request"""
SID_AUTH_RESPONSE = 1
"""SID for authentification response"""
DID_AUTH_SEED = 0
"""DID for authentification (seed)"""
DID_AUTH_KEY = 1
"""DID for authentification (key)"""
SID_CHANNEL_NAME_REQUEST = 8
"""SID for channel name exchange request """
SID_CHANNEL_NAME_RESPONSE = 9
"""SID for channel name exchange response"""
DID_CHANNEL_NAME = 0
"""DID for channel name """
SID_READ_REQUEST = 10
"""SID for a read data request"""
SID_READ_RESPONSE = 11
"""SID for read data response"""
SID_WRITE_REQUEST = 20
"""SID for a write data request"""
SID_WRITE_RESPONSE = 21
"""SID for a write data response"""
SID_EXECUTE_REQUEST = 30
"""SID for a execute request"""
SID_EXECUTE_RESPONSE = 31
"""SID for a execute response"""

STATUS_OKAY = 0
"""Status for 'okay'"""
STATUS_BUFFERING_UNHANDLED_REQUEST = 1
"""Status for 'unhandled request'"""
STATUS_CALLBACK_ERROR = 2
"""Status for 'callback errors'"""
STATUS_AUTH_REQUIRED = 3
"""Status for 'authentification is required'"""
STATUS_SERVICE_OR_DATA_UNKNOWN = 4
"""Status for 'service or data unknown'"""
STATUS_CHECKSUM_ERROR = 5
"""Status for 'checksum error'"""
STATUS_OPERATION_NOT_PERMITTED = 6
"""Status for 'operation not permitted'"""
STATUS_LOG_LVL = {
    STATUS_OKAY: logging.INFO,
    STATUS_BUFFERING_UNHANDLED_REQUEST: logging.WARNING,
    STATUS_CALLBACK_ERROR: logging.ERROR,
    STATUS_AUTH_REQUIRED: logging.WARNING,
    STATUS_SERVICE_OR_DATA_UNKNOWN: logging.ERROR,
    STATUS_CHECKSUM_ERROR: logging.ERROR,
    STATUS_OPERATION_NOT_PERMITTED: logging.WARNING,
}
"""Status depending log level for messages"""

AUTH_STATE_UNTRUSTED_CONNECTION = 0
"""Authentification Status for an 'Untrusted Connection'"""
AUTH_STATE_SEED_REQUESTED = 1
"""Authentification Status for 'Seed was requested'"""
AUTH_STATE_SEED_TRANSFERRED = 2
"""Authentification Status for 'Seed has been sent'"""
AUTH_STATE_KEY_TRANSFERRED = 3
"""Authentification Status for 'Key has been sent'"""
AUTH_STATE_TRUSTED_CONNECTION = 4
"""Authentification Status for a 'Trusted Connection'"""
AUTH_STATE__NAMES = {AUTH_STATE_UNTRUSTED_CONNECTION: 'Untrusted Connection',
```

```
116                     AUTH_STATE_SEED_REQUESTED: 'Seed was requested',
117                     AUTH_STATE_SEED_TRANSFERRED: 'Seed has been sent',
118                     AUTH_STATE_KEY_TRANSFERRED: 'Key has been sent',
119                     AUTH_STATE_TRUSTED_CONNECTION: 'Trusted Connection'}
120 """Authentification Status names for previous defined authentification states"""
121
122
123 class RequestSidExistsError(Exception):
124     pass
125
126
127 class ResponseSidExistsError(Exception):
128     pass
129
130
131 class _callback_storage(dict):
132     DEFAULT_CHANNEL_NAME = 'all_others'
133
134     def __init__(self, channel_name, log_prefix):
135         self.init_channel_name(channel_name)
136         self.__log_prefix__ = log_prefix
137         dict.__init__(self)
138
139     def init_channel_name(self, channel_name):
140         if channel_name is None:
141             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + self.
    DEFAULT_CHANNEL_NAME)
142         else:
143             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' +
    channel_name)
144
145     def get(self, service_id, data_id):
146         if dict.get(self, service_id, {}).get(data_id, None) is not None:
147             return self[service_id][data_id]
148         elif dict.get(self, service_id, {}).get(None, None) is not None:
149             return self[service_id][None]
150         elif dict.get(self, None, {}).get(data_id, None) is not None:
151             return self[None][data_id]
152         elif dict.get(self, None, {}).get(None, None) is not None:
153             return self[None][None]
154         else:
155             return (None, None, None)
156
157     def add(self, service_id, data_id, callback, *args, **kwargs):
158         cb_data = self.get(service_id, data_id)
159         if dict.get(self, service_id, {}).get(data_id, None) is not None:
160             if callback is None:
161                 self.logger.warning("%s Deleting existing callback %s for service_id (%s) and
    data_id (%s)!",
162                                     self.__log_prefix__(), repr(cb_data[0].__name__), repr(
    service_id), repr(data_id))
163                 del (self[service_id][data_id])
164                 return
165             else:
166                 self.logger.warning("%s Overwriting existing callback %s for service_id (%s) and
    data_id (%s) to %s!",
167                                     self.__log_prefix__(), repr(cb_data[0].__name__), repr(
    service_id), repr(data_id), repr(callback.__name__))
168         else:
169             self.logger.debug("%s Adding callback %s for SID=%s and DID=%s", self.__log_prefix__
    (),
170                               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 
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 optinal 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 authentification,
        otherwise you need to do it manually, if needed.
244     :type auto_auth: bool
245     :param channel_name: An optional parameter to set a channel name for logging of the
        communication.
246     :type channel_name: str
247
248     .. hint::
249
250         * The Service-ID is designed to identify the type of the communication (e.g. :const:`
        READ_REQUEST`, :const:`WRITE_REQUEST`, :const:`READ_RESPONSE`, :const:`WRITE_RESPONSE`, ...)
251         * The Data-ID is designed to identify the requests / responses using the same Service_ID.
252
253     .. note:: The :class:`comm_instance` needs to have at least the following interface:
254
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 authentification 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_example.py
273
274     Will result to the following output:
275
276     .. literalinclude:: socket_protocol/_examples_/socket_protocol_example.log
277     """
278     DEFAULT_CHANNEL_NAME = 'all_others'
279
280     def __init__(self, comm_instance, secret=None, auto_auth=False, channel_name=None):
281         self.__comm_inst__ = comm_instance
282         self.__secret__ = secret
283         self.__auto_auth__ = auto_auth
```

```
284         #
285         self.__auth_whitelist__ = {}
286         self.__sid_response_dict__ = {}
287         self.__sid_name_dict__ = {}
288         self.__did_name_dict__ = {}
289         #
290         self.__callbacks__ = _callback_storage(channel_name, self.__log_prefix__)
291         self.__init_channel_name__(channel_name)
292         #
293         self.__status_name_dict = {}
294         self.add_status(STATUS_OKAY, 'okay')
295         self.add_status(STATUS_BUFFERING_UNHANDLED_REQUEST, 'no callback for service, data
    buffered')
296         self.add_status(STATUS_CALLBACK_ERROR, 'callback error')
297         self.add_status(STATUS_AUTH_REQUIRED, 'authentification required')
298         self.add_status(STATUS_SERVICE_OR_DATA_UNKNOWN, 'service or data unknown')
299         self.add_status(STATUS_CHECKSUM_ERROR, 'checksum error')
300         self.add_status(STATUS_OPERATION_NOT_PERMITTED, 'operation not permitted')
301         #
302         self.__clean_receive_buffer__()
303
304         self.add_service(SID_AUTH_REQUEST, SID_AUTH_RESPONSE, 'authentification request', '
    authentification response')
305         self.add_data((SID_AUTH_REQUEST, SID_AUTH_RESPONSE), DID_AUTH_SEED, 'seed')
306         self.add_data(SID_AUTH_REQUEST, DID_AUTH_KEY, 'key')
307         self.add_data(SID_AUTH_RESPONSE, DID_AUTH_KEY, 'key')
308         self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_SEED)
309         self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_SEED)
310         self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_KEY)
311         self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_KEY)
312         self.__callbacks__.add(SID_AUTH_REQUEST, DID_AUTH_SEED, self.
    __authentificate_create_seed__)
313         self.__callbacks__.add(SID_AUTH_RESPONSE, DID_AUTH_SEED, self.
    __authentificate_create_key__)
314         self.__callbacks__.add(SID_AUTH_REQUEST, DID_AUTH_KEY, self.__authentificate_check_key__)
315         self.__callbacks__.add(SID_AUTH_RESPONSE, DID_AUTH_KEY, self.
    __authentificate_process_feedback__)
316         self.__authentification_state_reset__()
317
318         self.add_service(SID_CHANNEL_NAME_REQUEST, SID_CHANNEL_NAME_RESPONSE, 'channel name
    request', 'channel name response')
319         self.add_data((SID_CHANNEL_NAME_REQUEST, SID_CHANNEL_NAME_RESPONSE), DID_CHANNEL_NAME, '
    name')
320         self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME_REQUEST, DID_CHANNEL_NAME)
321         self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME_RESPONSE, DID_CHANNEL_NAME)
322         self.__callbacks__.add(SID_CHANNEL_NAME_REQUEST, DID_CHANNEL_NAME, self.
    __channel_name_request__)
323         self.__callbacks__.add(SID_CHANNEL_NAME_RESPONSE, DID_CHANNEL_NAME, self.
    __channel_name_response__)
324
325         self.add_service(SID_READ_REQUEST, SID_READ_RESPONSE, 'read data request', 'read data
    response')
326         self.add_service(SID_WRITE_REQUEST, SID_WRITE_RESPONSE, 'write data request', 'write data
     response')
327         self.add_service(SID_EXECUTE_REQUEST, SID_EXECUTE_RESPONSE, 'execute request', 'execute
    response')
328
329         self.__seed__ = None
330         self.__comm_inst__.register_callback(self.__data_available_callback__)
331         self.__comm_inst__.register_connect_callback(self.__connection_established__)
332         self.__comm_inst__.register_disconnect_callback(self.__authentification_state_reset__)
333         self.logger.info('%s Initialisation finished.', self.__log_prefix__())
```

```
334
335     def __analyse_frame__(self, frame):
336         if frame is not None:
337             return data_storage(json.loads(frame[:-4].decode('utf-8')))
338
339     def __authenticate_check_key__(self, msg):
340         key = msg.get_data()
341         if key == self.__authenticate_salt_and_hash__(self.__seed__):
342             self.__authentification_state__ = AUTH_STATE_TRUSTED_CONNECTION
343             return STATUS_OKAY, True
344         else:
345             self.__authentification_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
346             return STATUS_OKAY, False
347
348     def __authenticate_create_key__(self, msg):
349         self.__authentification_state__ = AUTH_STATE_KEY_TRANSFERRED
350         seed = msg.get_data()
351         key = self.__authenticate_salt_and_hash__(seed)
352         self.send(SID_AUTH_REQUEST, DID_AUTH_KEY, key)
353
354     def __authenticate_create_seed__(self, msg):
355         self.__authentification_state__ = AUTH_STATE_SEED_TRANSFERRED
356         self.__seed__ = binascii.hexlify(os.urandom(32)).decode('utf-8')
357         return STATUS_OKAY, self.__seed__
358
359     def __authenticate_process_feedback__(self, msg):
360         feedback = msg.get_data()
361         if feedback:
362             self.__authentification_state__ = AUTH_STATE_TRUSTED_CONNECTION
363             self.logger.info("%s Got positive authentification feedback", self.__log_prefix__())
364         else:
365             self.__authentification_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
366             self.logger.warning("%s Got negative authentification feedback", self.__log_prefix__
    ())
367         return STATUS_OKAY, None
368
369     def __authenticate_salt_and_hash__(self, seed):
370         return hashlib.sha512(bytes(seed, 'utf-8') + self.__secret__).hexdigest()
371
372     def __authentification_state_reset__(self):
373         self.logger.info("%s Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION"
    , self.__log_prefix__())
374         self.__authentification_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
375
376     def __authentification_required__(self, service_id, data_id):
377         return data_id not in self.__auth_whitelist__.get(service_id, [])
378
379     def __buffer_received_data__(self, msg):
380         if not msg.get_service_id() in self.__msg_buffer__:
381             self.__msg_buffer__[msg.get_service_id()] = {}
382         if not msg.get_data_id() in self.__msg_buffer__[msg.get_service_id()]:
383             self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()] = []
384         self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()].append(msg)
385         self.logger.debug("%s Message data is stored in buffer and is now ready to be retrieved
    by receive method", self.__log_prefix__())
386
387     def __build_frame__(self, msg):
388         data_frame = json.dumps(self.__mk_msg__(msg.get_status(), msg.get_service_id(), msg.
    get_data_id(), msg.get_data()))
389         data_frame = bytes(data_frame, 'utf-8')
390         checksum = self.__calc_chksum__(data_frame)
391         return data_frame + checksum
392
```

```
393     def __calc_chksum__(self, raw_data):
394         return struct.pack('>I', binascii.crc32(raw_data) & 0xffffffff)
395 
396     @property
397     def __channel_name__(self):
398         cn = self.logger.name.split('.')[-1]
399         if cn != self.DEFAULT_CHANNEL_NAME:
400             return cn
401 
402     def __channel_name_response__(self, msg):
403         data = msg.get_data()
404         if self.__channel_name__ is None and data is not None:
405             self.__init_channel_name__(data)
406             self.logger.info('%s channel name is now %s', self.__log_prefix__(), repr(self.
    __channel_name__))
407         return STATUS_OKAY, None
408 
409     def __channel_name_request__(self, msg):
410         data = msg.get_data()
411         if data is None:
412             return STATUS_OKAY, self.__channel_name__
413         else:
414             prev_channel_name = self.__channel_name__
415             self.__init_channel_name__(data)
416             if prev_channel_name is not None and prev_channel_name != data:
417                 self.logger.warning('%s overwriting user defined channel name from %s to %s',
418                                     self.__log_prefix__(), repr(prev_channel_name), repr(data))
419             elif prev_channel_name is None:
420                 self.logger.info('%s channel name is now %s', self.__log_prefix__(), repr(self.
    __channel_name__))
421             return STATUS_OKAY, None
422 
423     def __check_frame_checksum__(self, frame):
424         return self.__calc_chksum__(frame[:-4]) == frame[-4:]
425 
426     def __clean_receive_buffer__(self):
427         self.logger.debug("%s Cleaning up receive-buffer", self.__log_prefix__())
428         self.__msg_buffer__ = {}
429 
430     def __connection_established__(self):
431         self.__clean_receive_buffer__()
432         if self.__comm_inst__.IS_CLIENT:
433             self.send(SID_CHANNEL_NAME_REQUEST, 0, self.__channel_name__)
434         if self.__auto_auth__ and self.__comm_inst__.IS_CLIENT and self.__secret__ is not None:
435             self.authentificate()
436 
437     def __log_msg__(self, msg, rx_tx_prefix):
438         self.logger.log(
439             self.__status_log_lvl__(msg.get_status()),
440             '%s %s %s, %s, data: "%s"',
441             self.__log_prefix__(),
442             rx_tx_prefix,
443             self.__get_message_name__(msg.get_service_id(), msg.get_data_id()),
444             self.__get_status_name__(msg.get_status()),
445             repr(msg.get_data())
446         )
447 
448     def __data_available_callback__(self, comm_inst):
449         frame = comm_inst.receive()
450         msg = self.__analyse_frame__(frame)
451         if not self.__check_frame_checksum__(frame):
452             # Wrong Checksum
```

```
453             self.logger.log(self.__status_log_lvl__(STATUS_CHECKSUM_ERROR),
454                             "%s Received message has an invalid checksum. Message will be ignored.", self.__log_prefix__())
455             return  # No response needed
456         elif not self.check_authentification_state() and self.__authentification_required__(msg.get_service_id(), msg.get_data_id()):
457             # Authentification required
458             self.__log_msg__(msg, 'RX <-')
459             if msg.get_service_id() in self.__sid_response_dict__.keys():
460                 self.logger.log(self.__status_log_lvl__(STATUS_AUTH_REQUIRED),
461                                 "%s Authentification is required. Just sending negative response.", self.__log_prefix__())
462                 status = STATUS_AUTH_REQUIRED
463                 data = None
464             else:
465                 self.logger.log(self.__status_log_lvl__(STATUS_AUTH_REQUIRED),
466                                 "%s Authentification is required. Incomming message will be ignored.", self.__log_prefix__())
467                 return  # No response needed
468         else:
469             # Valid message
470             self.__log_msg__(msg, 'RX <-')
471             callback, args, kwargs = self.__callbacks__.get(msg.get_service_id(), msg.get_data_id())
472             if msg.get_service_id() in self.__sid_response_dict__.keys():
473                 #
474                 # REQUEST RECEIVED
475                 #
476                 if callback is None:
477                     self.logger.warning("%s Incomming message with no registered callback. Sending negative response.", self.__log_prefix__())
478                     status = STATUS_BUFFERING_UNHANDLED_REQUEST
479                     data = None
480                 else:
481                     self.logger.debug("%s Executing callback %s to process received data", self.__log_prefix__(), callback.__name__)
482                     try:
483                         status, data = callback(msg, *args, **kwargs)
484                     except Exception as e:
485                         self.logger.error('{lp} Exception raised. Check callback {callback_name}: "{message}" and it\'s return values for {msg_info}'.format(
486                             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())))
487                         status = STATUS_CALLBACK_ERROR
488                         data = None
489             else:
490                 #
491                 # RESPONSE RECEIVED
492                 #
493                 if callback is None:
494                     self.__buffer_received_data__(msg)
495                 else:
496                     self.logger.debug("%s Executing callback %s to process received data", self.__log_prefix__(), callback.__name__)
497                     try:
498                         callback(msg, *args, **kwargs)
499                     except Exception as e:
500                         self.logger.error('{lp} Exception raised. Check callback {callback_name}: "{message}" for {msg_info}'.format(lp=self.__log_prefix__(
501                         ), callback_name=callback.__name__, message=str(e), msg_info=self.__get_message_name__(msg.get_service_id(), msg.get_data_id())))
```

```
                return  # No response needed
        self.send(self.__sid_response_dict__[msg.get_service_id()], msg.get_data_id(), data,
    status=status)

    def __get_message_name__(self, service_id, data_id):
        return 'service: %s, data id: %s' % (
            self.__sid_name_dict__.get(service_id, repr(service_id)),
            self.__did_name_dict__.get(service_id, {}).get(data_id, repr(data_id)),
        )

    def __get_status_name__(self, status):
        return 'status: %s' % (self.__status_name_dict.get(status, 'unknown status: %s' % repr(
    status)))

    def __init_channel_name__(self, channel_name):
        self.__comm_inst__.init_channel_name(channel_name)
        self.__callbacks__.init_channel_name(channel_name)
        if channel_name is None:
            self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + self.
    DEFAULT_CHANNEL_NAME)
        else:
            self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' +
    channel_name)

    def __log_prefix__(self):
        return 'prot-client:' if self.__comm_inst__.IS_CLIENT else 'prot-server:'

    def __mk_msg__(self, status, service_id, data_id, data):
        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})

    def __status_log_lvl__(self, status):
        return STATUS_LOG_LVL.get(status, logging.CRITICAL)

    def add_data(self, service_id, data_id, name):
        """
        Method to add a name for a specific message.

        :param service_id: The Service-ID of the message. See class definitions starting with ``
    SID_``.
        :type service_id: int or list of ints
        :param data_id: The Data-ID of the message.
        :type data_id: int
        :param name: The Name for the transfered message.
        :type name: str
        """
        try:
            iter(service_id)
        except Exception:
            service_id = (service_id, )

        for sid in service_id:
            if sid not in self.__did_name_dict__:
                self.__did_name_dict__[sid] = {}
            self.__did_name_dict__[sid][data_id] = name

    def add_msg_to_auth_whitelist(self, service_id, data_id):
        """
        Method to add a specific message to the list, where no authentification is required.

        :param service_id: The Service-ID of the message. See class definitions starting with ``
    SID_``.
        :type service_id: int
        :param data_id: The Data-ID of the message.
        :type data_id: int
        """
```

```
561         if service_id not in self.__auth_whitelist__:
562             self.__auth_whitelist__[service_id] = []
563         self.__auth_whitelist__[service_id].append(data_id)
564         self.logger.debug('%s Adding Message (%s) to the authentification whitelist',
565                           self.__log_prefix__(), self.__get_message_name__(service_id, data_id))
566 
567     def add_service(self, req_sid, resp_sid, req_name=None, resp_name=None):
568         """
569         Method to add a Service defined by Request- and Response Serivce-ID.
570 
571         :param req_sid: The Request Service-ID.
572         :type req_sid: int
573         :param resp_sid: The Response Service-ID.
574         :type resp_sid: int
575         """
576         if req_sid in self.__sid_response_dict__:
577             self.logger.error('%s Service with Request-SID=%d and Response-SID=%d not added,
    because request SID is already registered',
578                               self.__log_prefix__(), req_sid, resp_sid)
579             raise RequestSidExistsError("Request for this Service is already registered")
580         elif resp_sid in self.__sid_response_dict__.values():
581             self.logger.error('%s Service with Request-SID=%d and Response-SID=%d not added,
    because response SID is already registered',
582                               self.__log_prefix__(), req_sid, resp_sid)
583             raise ResponseSidExistsError("Response for this Service is already registered")
584         else:
585             self.__sid_response_dict__[req_sid] = resp_sid
586             if req_name is not None:
587                 self.__sid_name_dict__[req_sid] = req_name
588             if resp_name is not None:
589                 self.__sid_name_dict__[resp_sid] = resp_name
590             self.logger.debug('%s Adding Service with Request=%s and Response=%s', self.
    __log_prefix__(),
591                               req_name or repr(req_sid), resp_name or repr(resp_sid))
592 
593     def add_status(self, status, name):
594         """
595         Method to add a name for a status.
596 
597         :param status: The Status. See class definitions starting with ``STATUS_``.
598         :type status: int
599         :param name: The Name for the Status.
600         :type name: str
601         """
602         self.__status_name_dict[status] = name
603 
604     def authentificate(self, timeout=2):
605         """
606         This method authetificates the client at the server.
607 
608         :param timeout: The timeout for the authentification (requesting seed, sending key and
    getting authentification_feedback).
609         :type timeout: float
610         :returns: True, if authentification was successfull; False, if not.
611         :rtype: bool
612 
613         .. note:: An authentification will only processed, if a secret had been given on
    initialisation.
614 
615         .. note:: Client and Server needs to use the same secret.
616         """
```

```
617         if self.__secret__ is not None:
618             self.__authentification_state__ = AUTH_STATE_SEED_REQUESTED
619             self.send(SID_AUTH_REQUEST, DID_AUTH_SEED, None)
620             cnt = 0
621             while cnt < timeout * 10:
622                 time.sleep(0.1)
623                 if self.__authentification_state__ == AUTH_STATE_TRUSTED_CONNECTION:
624                     return True
625                 elif self.__authentification_state__ == AUTH_STATE_UNTRUSTED_CONNECTION:
626                     break
627                 cnt += 1
628         return False
629 
630     def check_authentification_state(self):
631         """
632         This Method return the Authitification State as boolean value.
633 
634         :return: True, if authentification state is okay, otherwise False
635         :rtype: bool
636         """
637         return self.__secret__ is None or self.__authentification_state__ ==
    AUTH_STATE_TRUSTED_CONNECTION
638 
639     def connection_established(self):
640         """
641         This Method returns the Connection state including authentification as a boolean value.
642 
643         :return: True, if the connection is established (incl. authentification, if a secret has
    been given)
644         :rtype: bool
645         """
646         return self.is_connected() and (self.__secret__ is None or self.
    check_authentification_state())
647 
648     def is_connected(self):
649         """
650         This Methods returns Connection state of the Communication Instance :func:`comm_instance.
    is_connected`.
651 
652         :return: True if the :class:`comm_instance` is connected, otherwise False..
653         :rtype: bool
654         """
655         return self.__comm_inst__.is_connected()
656 
657     def receive(self, service_id, data_id, timeout=1):
658         """
659         This Method returns a message object for a defined message or None, if this message is
    not available after the given timout.
660 
661         :param service_id: The Service-ID for the message. See class definitions starting with ``
    SID_``.
662         :type service_id: int
663         :param data_id: The Data-ID for the message.
664         :type data_id: int
665         :param timeout: The timeout for receiving.
666         :type timeout: float
667         :returns: The received data storage object or None, if no data was received.
668         :rtype: data_storage
669         """
```

```
670         data = None
671         cnt = 0
672         while data is None and cnt < timeout * 10:
673             try:
674                 data = self.__msg_buffer__.get(service_id, {}).get(data_id, []).pop(0)
675             except IndexError:
676                 data = None
677             cnt += 1
678             time.sleep(0.1)
679         if data is None and cnt >= timeout * 10:
680             self.logger.warning('%s TIMEOUT (%ss): Requested data (service_id: %s; data_id: %s) not in buffer.',
681                                 self.__log_prefix__(), repr(timeout), repr(service_id), repr(data_id))
682         return data
683 
684     def reconnect(self):
685         """
686         This methods initiates a reconnect by calling :func:`comm_instance.reconnect`.
687         """
688         return self.__comm_inst__.reconnect()
689 
690     def register_callback(self, service_id, data_id, callback, *args, **kwargs):
691         """
692         This method registers a callback for the given parameters. Giving ``None`` means, that all Service-IDs or all Data-IDs are used.
693         If a message hitting these parameters has been received, the callback will be executed.
694 
695         :param service_id: The Service-ID for the message. See class definitions starting with ``SID_``.
696         :type service_id: int
697         :param data_id: The Data-ID for the message.
698         :type data_id: int
699 
700         .. note:: The :func:`callback` is priorised in the following order:
701 
702             * Callbacks with defined Service-ID and Data-ID.
703             * Callbacks with a defined Service-ID and all Data-IDs.
704             * Callbacks with a defined Data-ID and all Service-IDs.
705             * Unspecific Callbacks.
706 
707         .. note:: The :func:`callback` is executed with these arguments:
708 
709             **Parameters given at the callback call:**
710 
711             * The first Arguments is the received message as :class:`data_storage` object.
712             * Further arguments given at registration.
713             * Further keyword arguments given at registration.
714 
715             **Return value of the callback:**
716 
717             If the Callback is a Request Callback for a registered Service, the return value has to be a tuple or list with
718 
719             * :const:`response_status`: The response status (see class definitions starting with :const:`STA_*`.
720             * :const:`response_data`: A JSON iterable object to be used as data for the response.
721 
722         .. note:: Only registered services will respond via the callbacks return values with the same data_id.
723         """
```

```
724         self.__callbacks__.add(service_id, data_id, callback, *args, **kwargs)
725
726     def send(self, service_id, data_id, data, status=STATUS_OKAY, timeout=2):
727         """
728         This methods sends out a message with the given content.
729
730         :param service_id: The Service-ID for the message. See class definitions starting with ``SERVICE_``.
731         :type service_id: int
732         :param data_id: The Data-ID for the message.
733         :type data_id: int
734         :param data: The data to be transfered. The data needs to be json compatible.
735         :type data: str
736         :param status: The Status for the message. All requests should have ``STATUS_OKAY``.
737         :type status: int
738         :param timeout: The timeout for sending data (e.g. time to establish new connection).
739         :type timeout: float
740         :return: True if data had been sent, otherwise False.
741         :rtype: bool
742         """
743         if (self.check_authentification_state() or not self.__authentification_required__(service_id, data_id)) or (service_id in self.__sid_response_dict__.values() and status == STATUS_AUTH_REQUIRED and data is None):
744             msg = data_storage(service_id=service_id, data_id=data_id, data=data, status=status)
745             self.__log_msg__(msg, 'TX ->')
746             return self.__comm_inst__.send(self.__build_frame__(msg), timeout=timeout)
747         else:
748             # Authentification required
749             self.logger.warning("%s Authentification is required. TX-Message %s, %s, data: %s will be ignored.", self.__log_prefix__(),
750                                 self.__get_message_name__(service_id, data_id), self.__get_status_name__(status), repr(data))
751             return False
752
753
754 class struct_json_protocol(pure_json_protocol):
755     """
756     This Class has the same functionality like :class:`pure_json_protocol`. The message length is less than for :class:`pure_json_protocol`, but the functionality and compatibility is reduced.
757     See also parent :py:class:`pure_json_protocol`.
758
759     .. note::
760         This class is depreceated and here for compatibility reasons (to support old clients or servers). Usage of :class:`pure_json_protocol` is recommended.
761     """
762
763     def __init__(self, *args, **kwargs):
764         pure_json_protocol.__init__(self, *args, **kwargs)
765
766     def __analyse_frame__(self, frame):
767         status, service_id, data_id = struct.unpack('>III', frame[0:12])
768         data = json.loads(frame[12:-1].decode('utf-8'))
769         return self.__mk_msg__(status, service_id, data_id, data)
770
771     def __build_frame__(self, msg):
772         frame = struct.pack('>III', msg.get_status(), msg.get_service_id(), msg.get_data_id())
773         frame += bytes(json.dumps(msg.get_data()), 'utf-8')
774         frame += self.__calc_chksum__(frame)
775         return frame
776
```

```
777     def __calc_chksum__(self, raw_data):
778         chksum = 0
779         for b in raw_data:
780             chksum ^= b
781         return bytes([chksum])
782 
783     def __check_frame_checksum__(self, frame):
784         return self.__calc_chksum__(frame[:-1]) == frame[-1:]
```