Unittest for `socket_protocol`

March 23, 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	af40b656c85d25ecf7eb1ee930b00fdf
Dependencies	
stringtools	09b4d1c41b828c8d1ccb723fa1fd79a9

1.2 Unittest Information

Unittest Information	
Version	e2f899a8597d10f58dc24606f3b9dcfe
Testruns with	python 3.11.2 (final)

1.3 Test System Information

System Information	
Architecture	64bit
Distribution	Debian GNU/Linux 12 bookworm
Hostname	ahorn
Kernel	6.1.0-17-amd64 (#1 SMP PREEMPT_DYNAMIC Debian 6.1.69-1 (2023-12-30))
Machine	x86_64
Path	/home/dirk/my_repositories/unittest/socket_protocol
System	Linux
Username	dirk

2 Statistic

2.1 Test-Statistic for testrun with python 3.11.2 (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	19.313s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:46:56,151
Finished-Time:	2025-03-23 16:46:56,151
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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:46:56,152
Finished-Time:	2025-03-23 16:46:56,152
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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:46:56,152
Finished-Time:	2025-03-23 16:46:56,153
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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:46:56,153
Finished-Time:	2025-03-23 16:46:56,153
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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:46:56,153
Finished-Time:	2025-03-23 16:46:56,907
Time-Consumption	0.754s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:46:56,908
Finished-Time:	2025-03-23 16:46:57,870
Time-Consumption	0.962s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:46:57,871
Finished-Time:	2025-03-23 16:46:58,842
Time-Consumption	0.971s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:46:58,843
Finished-Time:	2025-03-23 16:47:01,557
Time-Consumption	2.714s
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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:01,558
Finished-Time:	2025-03-23 16:47:03,540
Time-Consumption	1.981s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:03,541
Finished-Time:	2025-03-23 16:47:06,027
Time-Consumption	2.487s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:06,028
Finished-Time:	2025-03-23 16:47:07,782
Time-Consumption	1.754s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:07,783
Finished-Time:	2025-03-23 16:47:08,650
Time-Consumption	0.868s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:08,651
Finished-Time:	2025-03-23 16:47:09,007
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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:09,008
Finished-Time:	2025-03-23 16:47:09,981
Time-Consumption	0.973s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:09,981
Finished-Time:	2025-03-23 16:47:10,542
Time-Consumption	0.560s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:10,543
Finished-Time:	2025-03-23 16:47:11,099
Time-Consumption	0.556s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:11,100
Finished-Time:	2025-03-23 16:47:11,662
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'>).

3.3.5 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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:11,663
Finished-Time:	2025-03-23 16:47:12,837
Time-Consumption	1.174s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:12,838
Finished-Time:	2025-03-23 16:47:13,653
Time-Consumption	0.815s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:13,653
Finished-Time:	2025-03-23 16:47:14,012
Time-Consumption	0.359s

Testsummary:

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

3.4.3 Reconnect Method

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:14,013
Finished-Time:	2025-03-23 16:47:14,712
Time-Consumption	0.699s

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 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/socket_protocol/unittest/src/report/__init__.py (327)
Start-Time:	2025-03-23 16:47:14,713
Finished-Time:	2025-03-23 16:47:15,478
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 3.11.2 (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:
↪  "'0110a97205e47154131da1a4f6026df6af13cce1f515b3c6dccffe91f0191b46'"
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 30 31 31 30
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 30 31 31 30
comm-server: TX -> (64): 61 39 37 32 30 35 65 34 37 31 35 34 31 33 31 64 61 31 61 34 66 36 30
↪  32 36 64 66 36 61 66 31 33 63 63 65 31 66 35 31 35 62 33 63 36 64 63 63 66 66 65 39 31 66
↪  30 31 39 31 62 34 36 22 7d 02 a6
comm-client: RX <- (64): 61 39 37 32 30 35 65 34 37 31 35 34 31 33 31 64 61 31 61 34 66 36 30
↪  32 36 64 66 36 61 66 31 33 63 63 65 31 66 35 31 35 62 33 63 36 64 63 63 66 66 65 39 31 66
↪  30 31 39 31 62 34 36 22 7d 02 a6
comm-server: TX -> (4): ff d8 3a 3e
comm-client: RX <- (4): ff d8 3a 3e
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'0110a97205e47154131da1a4f6026df6af13cce1f515b3c6dccffe91f0191b46'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'ccad1e0dbfa7138e8617021934623e859f9ec96e32bb9b42d4ae3c0576808ddfad5848fbb0b9277fab96617
↪  60f5acfba1f50b84f3aad5fd85dc7cdc69b35f179'"
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 63 63 61 64
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 63 63 61 64
```

```
comm-client: TX -> (64): 31 65 30 64 62 66 61 37 31 33 38 65 38 36 31 37 30 32 31 39 33 34 36
↪  32 33 65 38 35 39 66 39 65 63 39 36 65 33 32 62 62 39 62 34 32 64 34 61 65 33 63 30 35 37
↪  36 38 30 38 64 64 66 61 64 35 38
comm-server: RX <- (64): 31 65 30 64 62 66 61 37 31 33 38 65 38 36 31 37 30 32 31 39 33 34 36
↪  32 33 65 38 35 39 66 39 65 63 39 36 65 33 32 62 62 39 62 34 32 64 34 61 65 33 63 30 35 37
↪  36 38 30 38 64 64 66 61 64 35 38
comm-client: TX -> (64): 34 38 66 62 62 30 62 39 32 37 37 66 61 62 39 36 36 31 37 36 30 66 35
↪  61 63 66 62 61 31 66 35 30 62 38 34 66 33 61 61 64 35 66 64 38 35 64 63 37 63 64 63 36 39
↪  62 33 35 66 31 37 39 22 7d 9f 4f
comm-server: RX <- (64): 34 38 66 62 62 30 62 39 32 37 37 66 61 62 39 36 36 31 37 36 30 66 35
↪  61 63 66 62 61 31 66 35 30 62 38 34 66 33 61 61 64 35 66 64 38 35 64 63 37 63 64 63 36 39
↪  62 33 35 66 31 37 39 22 7d 9f 4f
comm-client: TX -> (4): 25 8d 3a 3e
comm-server: RX <- (4): 25 8d 3a 3e
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'ccad1e0dbfa7138e8617021934623e859f9ec96e32bb9b42d4ae3c0576808ddfad5848fbb0b9277fab96617
↪  60f5acfba1f50b84f3aad5fd85dc7cdc69b35f179'"
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:
↪  "'9cbb1defd4e0909a9d1471cd7e9b045b639ea75e19d3e5f66be7d91d7426bfbe'"
```

```
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 63 62 62
```

```
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 63 62 62
```

```
comm-server: TX -> (64): 31 64 65 66 64 34 65 30 39 30 39 61 39 64 31 34 37 31 63 64 37 65 39
↪  62 30 34 35 62 36 33 39 65 61 37 35 65 31 39 64 33 65 35 66 36 36 62 65 37 64 39 31 64 37
↪  34 32 36 62 66 62 65 22 7d 20 ad
```

```
comm-client: RX <- (64): 31 64 65 66 64 34 65 30 39 30 39 61 39 64 31 34 37 31 63 64 37 65 39
↪  62 30 34 35 62 36 33 39 65 61 37 35 65 31 39 64 33 65 35 66 36 36 62 65 37 64 39 31 64 37
↪  34 32 36 62 66 62 65 22 7d 20 ad
```

```
comm-server: TX -> (4): 12 5e 3a 3e
```

```
comm-client: RX <- (4): 12 5e 3a 3e
```

```
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'9cbb1defd4e0909a9d1471cd7e9b045b639ea75e19d3e5f66be7d91d7426bfbe'"
```

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

```
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'2fdf7fec8e11a6a11276d80d8a191e340fee1d7e7212d633ea4b378a965a467530c43e2570b51c30b8a68b5
↪  1f4c3d9b0f4c25d3cbb91ed1b8130bafabaf226ce'"
```

```
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 32 66 64 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 32 66 64 66
```

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

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

```
comm-client: TX -> (64): 33 65 32 35 37 30 62 35 31 63 33 30 62 38 61 36 38 62 35 31 66 34 63
↪  33 64 39 62 30 66 34 63 32 35 64 33 63 62 62 39 31 65 64 31 62 38 31 33 30 62 61 66 61 62
↪  61 66 32 32 36 63 65 22 7d 11 cd
```

```
comm-server: RX <- (64): 33 65 32 35 37 30 62 35 31 63 33 30 62 38 61 36 38 62 35 31 66 34 63
↪  33 64 39 62 30 66 34 63 32 35 64 33 63 62 62 39 31 65 64 31 62 38 31 33 30 62 61 66 61 62
↪  61 66 32 32 36 63 65 22 7d 11 cd
```

```
comm-client: TX -> (4): ed d9 3a 3e
```

```
comm-server: RX <- (4): ed d9 3a 3e
```

```
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'2fdf7fec8e11a6a11276d80d8a191e340fee1d7e7212d633ea4b378a965a467530c43e2570b51c30b8a68b5
↪  1f4c3d9b0f4c25d3cbb91ed1b8130bafabaf226ce'"
```

```
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:
↪  "'a37202c5ecdf82421bae20184dfadda1847aeca475e1766a54415ac866f4a081'"
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 61 33 37 32
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪  69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪  74 61 22 3a 3d 20 22 61 33 37 32
comm-server: TX -> (64): 30 32 63 35 65 63 64 66 38 32 34 32 31 62 61 65 32 30 31 38 34 64 66
↪  61 64 64 61 31 38 34 37 61 65 63 61 34 37 35 65 31 37 36 36 61 35 34 34 31 35 61 63 38 36
↪  36 66 34 61 30 38 31 22 7d 96 4c
comm-client: RX <- (64): 30 32 63 35 65 63 64 66 38 32 34 32 31 62 61 65 32 30 31 38 34 64 66
↪  61 64 64 61 31 38 34 37 61 65 63 61 34 37 35 65 31 37 36 36 61 35 34 34 31 35 61 63 38 36
↪  36 66 34 61 30 38 31 22 7d 96 4c
comm-server: TX -> (4): 2a 7a 3a 3e
comm-client: RX <- (4): 2a 7a 3a 3e
```

```
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'a37202c5ecdf82421bae20184dfadda1847aeca475e1766a54415ac866f4a081'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'2b07305f68bfcd4c41489d267e614a33eaa379f93b045bcd2db549c61415c3d354286ec1e890088280a2f57
↪  389b3df14e96c3bf69fe2a8f4091211b385f8d6e7'"
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 32 62 30 37
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 32 62 30 37
comm-client: TX -> (64): 33 30 35 66 36 38 62 66 63 64 34 63 34 31 34 38 39 64 32 36 37 65 36
↪  31 34 61 33 33 65 61 61 33 37 39 66 39 33 62 30 34 35 62 63 64 32 64 62 35 34 39 63 36 31
↪  34 31 35 63 33 64 33 35 34 32 38
comm-server: RX <- (64): 33 30 35 66 36 38 62 66 63 64 34 63 34 31 34 38 39 64 32 36 37 65 36
↪  31 34 61 33 33 65 61 61 33 37 39 66 39 33 62 30 34 35 62 63 64 32 64 62 35 34 39 63 36 31
↪  34 31 35 63 33 64 33 35 34 32 38
comm-client: TX -> (64): 36 65 63 31 65 38 39 30 30 38 38 32 38 30 61 32 66 35 37 33 38 39 62
↪  33 64 66 31 34 65 39 36 63 33 62 66 36 39 66 65 32 61 38 66 34 30 39 31 32 31 31 62 33 38
↪  35 66 38 64 36 65 37 22 7d 77 f7
comm-server: RX <- (64): 36 65 63 31 65 38 39 30 30 38 38 32 38 30 61 32 66 35 37 33 38 39 62
↪  33 64 66 31 34 65 39 36 63 33 62 66 36 39 66 65 32 61 38 66 34 30 39 31 32 31 31 62 33 38
↪  35 66 38 64 36 65 37 22 7d 77 f7
comm-client: TX -> (4): 1d 1c 3a 3e
comm-server: RX <- (4): 1d 1c 3a 3e
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'2b07305f68bfcd4c41489d267e614a33eaa379f93b045bcd2db549c61415c3d354286ec1e890088280a2f57
↪  389b3df14e96c3bf69fe2a8f4091211b385f8d6e7'"
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:
↪  "'0bd0e2cc2069395cebe88c192a41001e87e98f0e95c99ff45eb1806ccdf78aee'"
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 30 62 64 30
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 30 62 64 30
comm-server: TX -> (64): 65 32 63 63 32 30 36 39 33 39 35 63 65 62 65 38 38 63 31 39 32 61 34
↪  31 30 30 31 65 38 37 65 39 38 66 30 65 39 35 63 39 39 66 66 34 35 65 62 31 38 30 36 63 63
↪  64 66 37 38 61 65 65 22 7d 08 ee
comm-client: RX <- (64): 65 32 63 63 32 30 36 39 33 39 35 63 65 62 65 38 38 63 31 39 32 61 34
↪  31 30 30 31 65 38 37 65 39 38 66 30 65 39 35 63 39 39 66 66 34 35 65 62 31 38 30 36 63 63
↪  64 66 37 38 61 65 65 22 7d 08 ee
comm-server: TX -> (4): 07 8b 3a 3e
comm-client: RX <- (4): 07 8b 3a 3e
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'0bd0e2cc2069395cebe88c192a41001e87e98f0e95c99ff45eb1806ccdf78aee'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'22df2547b66672daa68554babb89a4144f4edf8a0bb6c6e1ea7124567b203e2edf962715a1a3ca50de2e63d
↪  dff2e055883bd800936f76a935b7d68c14f6c9c3c'"
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 32 32 64 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 32 32 64 66
```

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

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

```
comm-client: TX -> (64): 32 37 31 35 61 31 61 33 63 61 35 30 64 65 32 65 36 33 64 64 66 66 32
↪  65 30 35 35 38 38 33 62 64 38 30 30 39 33 36 66 37 36 61 39 33 35 62 37 64 36 38 63 31 34
↪  66 36 63 39 63 33 63 22 7d a8 eb
```

```
comm-server: RX <- (64): 32 37 31 35 61 31 61 33 63 61 35 30 64 65 32 65 36 33 64 64 66 66 32
↪  65 30 35 35 38 38 33 62 64 38 30 30 39 33 36 66 37 36 61 39 33 35 62 37 64 36 38 63 31 34
↪  66 36 63 39 63 33 63 22 7d a8 eb
```

```
comm-client: TX -> (4): 03 85 3a 3e
```

```
comm-server: RX <- (4): 03 85 3a 3e
```

```
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'22df2547b66672daa68554babb89a4144f4edf8a0bb6c6e1ea7124567b203e2edf962715a1a3ca50de2e63d
↪  dff2e055883bd800936f76a935b7d68c14f6c9c3c'"
```

```
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:
↪  "'33e315121cf78eb3d5b6aa11d62eb26c7e408540a15faaabbc054e34e3939a65'"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪  69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪  74 61 22 3a 3d 20 22 33 33 65 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 33 33 65 33
comm-server: TX -> (64): 31 35 31 32 31 63 66 37 38 65 62 33 64 35 62 36 61 61 31 31 64 36 32
↪  65 62 32 36 63 37 65 34 30 38 35 34 30 61 31 35 66 61 61 61 62 62 63 30 35 34 65 33 34 65
↪  33 39 33 39 61 36 35 22 7d 32 9c
comm-client: RX <- (64): 31 35 31 32 31 63 66 37 38 65 62 33 64 35 62 36 61 61 31 31 64 36 32
↪  65 62 32 36 63 37 65 34 30 38 35 34 30 61 31 35 66 61 61 61 62 62 63 30 35 34 65 33 34 65
↪  33 39 33 39 61 36 35 22 7d 32 9c
comm-server: TX -> (4): f4 6d 3a 3e
comm-client: RX <- (4): f4 6d 3a 3e
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'33e315121cf78eb3d5b6aa11d62eb26c7e408540a15faaabbc054e34e3939a65'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'62b46adfaf683dc8157251a2010826f6b709b2e902d90727502e0dbd05e55c453ce76973fe25f49bbdf2ca0
↪  9d52db3b247adec86000ec2536fe5a928d4dcd635'"
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 36 32 62 34
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 36 32 62 34
comm-client: TX -> (64): 36 61 64 66 61 66 36 38 33 64 63 38 31 35 37 32 35 31 61 32 30 31 30
↪  38 32 36 66 36 62 37 30 39 62 32 65 39 30 32 64 39 30 37 32 37 35 30 32 65 30 64 62 64 30
↪  35 65 35 35 63 34 35 33 63 65 37
comm-server: RX <- (64): 36 61 64 66 61 66 36 38 33 64 63 38 31 35 37 32 35 31 61 32 30 31 30
↪  38 32 36 66 36 62 37 30 39 62 32 65 39 30 32 64 39 30 37 32 37 35 30 32 65 30 64 62 64 30
↪  35 65 35 35 63 34 35 33 63 65 37
comm-client: TX -> (64): 36 39 37 33 66 65 32 35 66 34 39 62 62 64 66 32 63 61 30 39 64 35 32
↪  64 62 33 62 32 34 37 61 64 65 63 38 36 30 30 30 65 63 32 35 33 36 66 65 35 61 39 32 38 64
↪  34 64 63 64 36 33 35 22 7d eb 3c
comm-server: RX <- (64): 36 39 37 33 66 65 32 35 66 34 39 62 62 64 66 32 63 61 30 39 64 35 32
↪  64 62 33 62 32 34 37 61 64 65 63 38 36 30 30 30 65 63 32 35 33 36 66 65 35 61 39 32 38 64
↪  34 64 63 64 36 33 35 22 7d eb 3c
comm-client: TX -> (4): fa 73 3a 3e
comm-server: RX <- (4): fa 73 3a 3e
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'62b46adfaf683dc8157251a2010826f6b709b2e902d90727502e0dbd05e55c453ce76973fe25f49bbdf2ca0
↪  9d52db3b247adec86000ec2536fe5a928d4dcd635'"
```

```
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-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (21): 3a 3c 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13 3a 3e
comm-server: RX <- (21): 3a 3c 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13 3a 3e
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:**

```

```
24          See also the :download:`unittest <socket_protocol/_testresults_/unittest.pdf>`
        documentation.
25
26  **Module Documentation:**
27
28  """
29  __DEPENDENCIES__ = ['stringtools']
30
31  import stringtools
32
33  import binascii
34  import hashlib
35  import json
36  import logging
37  import os
38  import struct
39  import sys
40  import time
41
42
43  try:
44      from config import APP_NAME as ROOT_LOGGER_NAME
45  except ImportError:
46      ROOT_LOGGER_NAME = 'root'
47  logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)
48
49
50  __DESCRIPTION__ = """The Module {\\tt %s} is designed for point to point communication for client
        -server issues.
51  For more Information read the sphinx documentation.""" % __name__.replace('_', '\_')
52  """The Module Description"""
53  __INTERPRETER__ = (3, )
54  """The Tested Interpreter-Versions"""
55
56  SID_AUTH_REQUEST = 0
57  """SID for authentification request"""
58  SID_AUTH_RESPONSE = 1
59  """SID for authentification response"""
60  DID_AUTH_SEED = 0
61  """DID for authentification (seed)"""
62  DID_AUTH_KEY = 1
63  """DID for authentification (key)"""
64  SID_CHANNEL_NAME_REQUEST = 8
65  """SID for channel name exchange request """
66  SID_CHANNEL_NAME_RESPONSE = 9
67  """SID for channel name exchange response"""
68  DID_CHANNEL_NAME = 0
69  """DID for channel name """
70  SID_READ_REQUEST = 10
71  """SID for a read data request"""
72  SID_READ_RESPONSE = 11
73  """SID for read data response"""
74  SID_WRITE_REQUEST = 20
75  """SID for a write data request"""
76  SID_WRITE_RESPONSE = 21
77  """SID for a write data response"""
78  SID_EXECUTE_REQUEST = 30
79  """SID for a execute request"""
80  SID_EXECUTE_RESPONSE = 31
81  """SID for a execute response"""
82
```

```
83  STATUS_OKAY = 0
84  """Status for 'okay'"""
85  STATUS_BUFFERING_UNHANDLED_REQUEST = 1
86  """Status for 'unhandled request'"""
87  STATUS_CALLBACK_ERROR = 2
88  """Status for 'callback errors'"""
89  STATUS_AUTH_REQUIRED = 3
90  """Status for 'authentification is required'"""
91  STATUS_SERVICE_OR_DATA_UNKNOWN = 4
92  """Status for 'service or data unknown'"""
93  STATUS_CHECKSUM_ERROR = 5
94  """Status for 'checksum error'"""
95  STATUS_OPERATION_NOT_PERMITTED = 6
96  """Status for 'operation not permitted'"""
97  STATUS_LOG_LVL = {
98      STATUS_OKAY: logging.INFO,
99      STATUS_BUFFERING_UNHANDLED_REQUEST: logging.WARNING,
100     STATUS_CALLBACK_ERROR: logging.ERROR,
101     STATUS_AUTH_REQUIRED: logging.WARNING,
102     STATUS_SERVICE_OR_DATA_UNKNOWN: logging.ERROR,
103     STATUS_CHECKSUM_ERROR: logging.ERROR,
104     STATUS_OPERATION_NOT_PERMITTED: logging.WARNING,
105 }
106 """Status depending log level for messages"""
107
108 AUTH_STATE_UNTRUSTED_CONNECTION = 0
109 """Authentification Status for an 'Untrusted Connection'"""
110 AUTH_STATE_SEED_REQUESTED = 1
111 """Authentification Status for 'Seed was requested'"""
112 AUTH_STATE_SEED_TRANSFERRED = 2
113 """Authentification Status for 'Seed has been sent'"""
114 AUTH_STATE_KEY_TRANSFERRED = 3
115 """Authentification Status for 'Key has been sent'"""
116 AUTH_STATE_TRUSTED_CONNECTION = 4
117 """Authentification Status for a 'Trusted Connection'"""
118 AUTH_STATE__NAMES = {AUTH_STATE_UNTRUSTED_CONNECTION: 'Untrusted Connection',
119                      AUTH_STATE_SEED_REQUESTED: 'Seed was requested',
120                      AUTH_STATE_SEED_TRANSFERRED: 'Seed has been sent',
121                      AUTH_STATE_KEY_TRANSFERRED: 'Key has been sent',
122                      AUTH_STATE_TRUSTED_CONNECTION: 'Trusted Connection'}
123 """Authentification Status names for previous defined authentification states"""
124
125
126 class RequestSidExistsError(Exception):
127     pass
128
129
130 class ResponseSidExistsError(Exception):
131     pass
132
133
134 class _callback_storage(dict):
135     DEFAULT_CHANNEL_NAME = 'all_others'
136
137     def __init__(self, channel_name, log_prefix):
138         self.init_channel_name(channel_name)
139         self.__log_prefix__ = log_prefix
140         dict.__init__(self)
141
142     def init_channel_name(self, channel_name):
143         if channel_name is None:
144             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + self.
    DEFAULT_CHANNEL_NAME)
```

```
145         else:
146             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' +
    channel_name)
147 
148     def get(self, service_id, data_id):
149         if dict.get(self, service_id, {}).get(data_id, None) is not None:
150             return self[service_id][data_id]
151         elif dict.get(self, service_id, {}).get(None, None) is not None:
152             return self[service_id][None]
153         elif dict.get(self, None, {}).get(data_id, None) is not None:
154             return self[None][data_id]
155         elif dict.get(self, None, {}).get(None, None) is not None:
156             return self[None][None]
157         else:
158             return (None, None, None)
159 
160     def add(self, service_id, data_id, callback, *args, **kwargs):
161         cb_data = self.get(service_id, data_id)
162         if dict.get(self, service_id, {}).get(data_id, None) is not None:
163             if callback is None:
164                 self.logger.warning("%s Deleting existing callback %s for service_id (%s) and
    data_id (%s)!",
165                                     self.__log_prefix__(), repr(cb_data[0].__name__), repr(
    service_id), repr(data_id))
166                 del (self[service_id][data_id])
167                 return
168             else:
169                 self.logger.warning("%s Overwriting existing callback %s for service_id (%s) and
    data_id (%s) to %s!",
170                                     self.__log_prefix__(), repr(cb_data[0].__name__), repr(
    service_id), repr(data_id), repr(callback.__name__))
171         else:
172             self.logger.debug("%s Adding callback %s for SID=%s and DID=%s", self.__log_prefix__
    (),
173                               repr(callback.__name__), repr(service_id), repr(data_id))
174         if service_id not in self:
175             self[service_id] = {}
176         self[service_id][data_id] = (callback, args, kwargs)
177 
178 
179 class data_storage(dict):
180     """
181     This is a storage object for socket_protocol messages.
182 
183     :param status: The message status.
184     :type status: int
185     :param service_id: The Service-ID.
186     :type service_id: int
187     :param data_id: The Data-ID.
188     :type data_id: int
189     :param data: The transfered data.
190     :type data: any
191     """
192 
193     KEY_STATUS = 'status'
194     KEY_SERVICE_ID = 'service_id'
195     KEY_DATA_ID = 'data_id'
196     KEY_DATA = 'data'
197     ALL_KEYS = [KEY_DATA, KEY_DATA_ID, KEY_SERVICE_ID, KEY_STATUS]
198 
```

```
199     def __init__(self, *args, **kwargs):
200         dict.__init__(self, *args, **kwargs)
201         for key in self.ALL_KEYS:
202             if key not in self:
203                 self[key] = None
204 
205     def get_status(self, default=None):
206         """
207         This Method returns the message status.
208 
209         :param default: The default value, if no data is available.
210         """
211         return self.get(self.KEY_STATUS, default)
212 
213     def get_service_id(self, default=None):
214         """
215         This Method returns the message Service-ID.
216 
217         :param default: The default value, if no data is available.
218         """
219         return self.get(self.KEY_SERVICE_ID, default)
220 
221     def get_data_id(self, default=None):
222         """
223         This Method returns the message Data-ID.
224 
225         :param default: The default value, if no data is available.
226         """
227         return self.get(self.KEY_DATA_ID, default)
228 
229     def get_data(self, default=None):
230         """
231         This Method returns the message data.
232 
233         :param default: The default value, if no data is available.
234         """
235         return self.get(self.KEY_DATA, default)
236 
237 
238 class pure_json_protocol(object):
239     """
240     This `class` supports to transfer a message and it's data.
241 
242     :param comm_instance: A communication instance.
243     :type comm_instance: instance
244     :param secret: An optinal secret (e.g. created by ``binascii.hexlify(os.urandom(24))``).
245     :type secret: str
246     :param auto_auth: An optional parameter to enable (True) automatic authentification,
    otherwise you need to do it manually, if needed.
247     :type auto_auth: bool
248     :param channel_name: An optional parameter to set a channel name for logging of the
    communication.
249     :type channel_name: str
250 
251     .. hint::
252 
253         * 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`, ...)
254         * The Data-ID is designed to identify the requests / responses using the same Service_ID.
255 
256     .. note:: The :class:`comm_instance` needs to have at least the following interface:
257 
```

```
258         * A Method :func:`comm_instance.init_channel_name` to set the channel name.
259         * A Constant :const:`comm_instance.IS_CLIENT` to identify that the :class:`comm_instance`
     is a client (True) or a server (False).
260         * A Method :func:`comm_instance.is_connected` to identify if the instance is connected (
    True) or not (False).
261         * A Method :func:`comm_instance.reconnect` to initiate a reconnect.
262         * A Method :func:`comm_instance.register_callback` to register a data available callback.
263         * A Method :func:`comm_instance.register_connect_callback` to register a connect callback
    .
264         * A Method :func:`comm_instance.register_disconnect_callback` to register a disconnect
    callback.
265         * A Method :func:`comm_instance.send` to send data via the :class:`comm_instance`.
266
267     .. 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.
268
269     .. note::  The :const:`channel_name`-exchange will be initiated by the client directly after
    the the connection is established.
270
271         * 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.
272
273     **Example:**
274
275     .. literalinclude:: socket_protocol/_examples_/socket_protocol_client.py
276
277     and
278
279     .. literalinclude:: socket_protocol/_examples_/socket_protocol_server.py
280
281     Will result to the following output:
282
283     .. literalinclude:: socket_protocol/_examples_/socket_protocol_client.log
284     """
285     DEFAULT_CHANNEL_NAME = 'all_others'
286
287     def __init__(self, comm_instance, secret=None, auto_auth=False, channel_name=None):
288         self.__comm_inst__ = comm_instance
289         self.__secret__ = secret
290         self.__auto_auth__ = auto_auth
291         #
292         self.__auth_whitelist__ = {}
293         self.__sid_response_dict__ = {}
294         self.__sid_name_dict__ = {}
295         self.__did_name_dict__ = {}
296         #
297         self.__callbacks__ = _callback_storage(channel_name, self.__log_prefix__)
298         self.__init_channel_name__(channel_name)
299         #
300         self.__status_name_dict = {}
301         self.add_status(STATUS_OKAY, 'okay')
302         self.add_status(STATUS_BUFFERING_UNHANDLED_REQUEST, 'no callback for service, data
    buffered')
303         self.add_status(STATUS_CALLBACK_ERROR, 'callback error')
304         self.add_status(STATUS_AUTH_REQUIRED, 'authentification required')
305         self.add_status(STATUS_SERVICE_OR_DATA_UNKNOWN, 'service or data unknown')
306         self.add_status(STATUS_CHECKSUM_ERROR, 'checksum error')
307         self.add_status(STATUS_OPERATION_NOT_PERMITTED, 'operation not permitted')
308         #
309         self.__clean_receive_buffer__()
310
```

```
311         self.add_service(SID_AUTH_REQUEST, SID_AUTH_RESPONSE, 'authentification request', '
    authentification response')
312         self.add_data((SID_AUTH_REQUEST, SID_AUTH_RESPONSE), DID_AUTH_SEED, 'seed')
313         self.add_data(SID_AUTH_REQUEST, DID_AUTH_KEY, 'key')
314         self.add_data(SID_AUTH_RESPONSE, DID_AUTH_KEY, 'key')
315         self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_SEED)
316         self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_SEED)
317         self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_KEY)
318         self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_KEY)
319         self.__callbacks__.add(SID_AUTH_REQUEST, DID_AUTH_SEED, self.
    __authentificate_create_seed__)
320         self.__callbacks__.add(SID_AUTH_RESPONSE, DID_AUTH_SEED, self.
    __authentificate_create_key__)
321         self.__callbacks__.add(SID_AUTH_REQUEST, DID_AUTH_KEY, self.__authentificate_check_key__)
322         self.__callbacks__.add(SID_AUTH_RESPONSE, DID_AUTH_KEY, self.
    __authentificate_process_feedback__)
323         self.__authentification_state_reset__()
324
325         self.add_service(SID_CHANNEL_NAME_REQUEST, SID_CHANNEL_NAME_RESPONSE, 'channel name
    request', 'channel name response')
326         self.add_data((SID_CHANNEL_NAME_REQUEST, SID_CHANNEL_NAME_RESPONSE), DID_CHANNEL_NAME, '
    name')
327         self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME_REQUEST, DID_CHANNEL_NAME)
328         self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME_RESPONSE, DID_CHANNEL_NAME)
329         self.__callbacks__.add(SID_CHANNEL_NAME_REQUEST, DID_CHANNEL_NAME, self.
    __channel_name_request__)
330         self.__callbacks__.add(SID_CHANNEL_NAME_RESPONSE, DID_CHANNEL_NAME, self.
    __channel_name_response__)
331
332         self.add_service(SID_READ_REQUEST, SID_READ_RESPONSE, 'read data request', 'read data
    response')
333         self.add_service(SID_WRITE_REQUEST, SID_WRITE_RESPONSE, 'write data request', 'write data
     response')
334         self.add_service(SID_EXECUTE_REQUEST, SID_EXECUTE_RESPONSE, 'execute request', 'execute
    response')
335
336         self.__seed__ = None
337         self.__comm_inst__.register_callback(self.__data_available_callback__)
338         self.__comm_inst__.register_connect_callback(self.__connection_established__)
339         self.__comm_inst__.register_disconnect_callback(self.__authentification_state_reset__)
340         self.logger.info('%s Initialisation finished.', self.__log_prefix__())
341
342     def __analyse_frame__(self, frame):
343         if frame is not None:
344             return data_storage(json.loads(frame[:-4].decode('utf-8')))
345
346     def __authentificate_check_key__(self, msg):
347         key = msg.get_data()
348         if key == self.__authentificate_salt_and_hash__(self.__seed__):
349             self.__authentification_state__ = AUTH_STATE_TRUSTED_CONNECTION
350             return STATUS_OKAY, True
351         else:
352             self.__authentification_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
353             return STATUS_OKAY, False
354
355     def __authentificate_create_key__(self, msg):
356         self.__authentification_state__ = AUTH_STATE_KEY_TRANSFERRED
357         seed = msg.get_data()
358         key = self.__authentificate_salt_and_hash__(seed)
359         self.send(SID_AUTH_REQUEST, DID_AUTH_KEY, key)
360
```

```
361     def __authentificate_create_seed__(self, msg):
362         self.__authentification_state__ = AUTH_STATE_SEED_TRANSFERRED
363         self.__seed__ = binascii.hexlify(os.urandom(32)).decode('utf-8')
364         return STATUS_OKAY, self.__seed__
365 
366     def __authentificate_process_feedback__(self, msg):
367         feedback = msg.get_data()
368         if feedback:
369             self.__authentification_state__ = AUTH_STATE_TRUSTED_CONNECTION
370             self.logger.info("%s Got positive authentification feedback", self.__log_prefix__())
371         else:
372             self.__authentification_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
373             self.logger.warning("%s Got negative authentification feedback", self.__log_prefix__
    ())
374         return STATUS_OKAY, None
375 
376     def __authentificate_salt_and_hash__(self, seed):
377         return hashlib.sha512(bytes(seed, 'utf-8') + self.__secret__).hexdigest()
378 
379     def __authentification_state_reset__(self):
380         self.logger.info("%s Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION"
    , self.__log_prefix__())
381         self.__authentification_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
382 
383     def __authentification_required__(self, service_id, data_id):
384         return data_id not in self.__auth_whitelist__.get(service_id, [])
385 
386     def __buffer_received_data__(self, msg):
387         if not msg.get_service_id() in self.__msg_buffer__:
388             self.__msg_buffer__[msg.get_service_id()] = {}
389         if not msg.get_data_id() in self.__msg_buffer__[msg.get_service_id()]:
390             self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()] = []
391         self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()].append(msg)
392         self.logger.debug("%s Message data is stored in buffer and is now ready to be retrieved
    by receive method", self.__log_prefix__())
393 
394     def __build_frame__(self, msg):
395         data_frame = json.dumps(self.__mk_msg__(msg.get_status(), msg.get_service_id(), msg.
    get_data_id(), msg.get_data()))
396         data_frame = bytes(data_frame, 'utf-8')
397         checksum = self.__calc_chksum__(data_frame)
398         return data_frame + checksum
399 
400     def __calc_chksum__(self, raw_data):
401         return struct.pack('>I', binascii.crc32(raw_data) & 0xffffffff)
402 
403     @property
404     def __channel_name__(self):
405         cn = self.logger.name.split('.')[-1]
406         if cn != self.DEFAULT_CHANNEL_NAME:
407             return cn
408 
409     def __channel_name_response__(self, msg):
410         data = msg.get_data()
411         if self.__channel_name__ is None and data is not None:
412             self.__init_channel_name__(data)
413             self.logger.info('%s channel name is now %s', self.__log_prefix__(), repr(self.
    __channel_name__))
414         return STATUS_OKAY, None
415 
```

```
416     def __channel_name_request__(self, msg):
417         data = msg.get_data()
418         if data is None:
419             return STATUS_OKAY, self.__channel_name__
420         else:
421             prev_channel_name = self.__channel_name__
422             self.__init_channel_name__(data)
423             if prev_channel_name is not None and prev_channel_name != data:
424                 self.logger.warning('%s overwriting user defined channel name from %s to %s',
425                                     self.__log_prefix__(), repr(prev_channel_name), repr(data))
426             elif prev_channel_name is None:
427                 self.logger.info('%s channel name is now %s', self.__log_prefix__(), repr(self.
    __channel_name__))
428             return STATUS_OKAY, None
429 
430     def __check_frame_checksum__(self, frame):
431         return self.__calc_chksum__(frame[:-4]) == frame[-4:]
432 
433     def __clean_receive_buffer__(self):
434         self.logger.debug("%s Cleaning up receive-buffer", self.__log_prefix__())
435         self.__msg_buffer__ = {}
436 
437     def __connection_established__(self):
438         self.__clean_receive_buffer__()
439         if self.__comm_inst__.IS_CLIENT:
440             self.send(SID_CHANNEL_NAME_REQUEST, 0, self.__channel_name__)
441         if self.__auto_auth__ and self.__comm_inst__.IS_CLIENT and self.__secret__ is not None:
442             self.authentificate()
443 
444     def __log_msg__(self, msg, rx_tx_prefix):
445         self.logger.log(
446             self.__status_log_lvl__(msg.get_status()),
447             '%s %s %s, %s, data: "%s"',
448             self.__log_prefix__(),
449             rx_tx_prefix,
450             self.__get_message_name__(msg.get_service_id(), msg.get_data_id()),
451             self.__get_status_name__(msg.get_status()),
452             repr(msg.get_data())
453         )
454 
455     def __data_available_callback__(self, comm_inst):
456         frame = comm_inst.receive()
457         msg = self.__analyse_frame__(frame)
458         if not self.__check_frame_checksum__(frame):
459             # Wrong Checksum
460             self.logger.log(self.__status_log_lvl__(STATUS_CHECKSUM_ERROR),
461                             "%s Received message has an invalid checksum. Message will be ignored
    .", self.__log_prefix__())
462             return  # No response needed
463         elif not self.check_authentification_state() and self.__authentification_required__(msg.
    get_service_id(), msg.get_data_id()):
464             # Authentification required
465             self.__log_msg__(msg, 'RX <-')
466             if msg.get_service_id() in self.__sid_response_dict__.keys():
467                 self.logger.log(self.__status_log_lvl__(STATUS_AUTH_REQUIRED),
468                                 "%s Authentification is required. Just sending negative response.
    ", self.__log_prefix__())
469                 status = STATUS_AUTH_REQUIRED
470                 data = None
471             else:
472                 self.logger.log(self.__status_log_lvl__(STATUS_AUTH_REQUIRED),
473                                 "%s Authentification is required. Incomming message will be
    ignored.", self.__log_prefix__())
```

```
474                 return  # No response needed
475         else:
476             # Valid message
477             self.__log_msg__(msg, 'RX <--')
478             callback, args, kwargs = self.__callbacks__.get(msg.get_service_id(), msg.get_data_id
    ())
479             if msg.get_service_id() in self.__sid_response_dict__.keys():
480                 #
481                 # REQUEST RECEIVED
482                 #
483                 if callback is None:
484                     self.logger.warning("%s Incomming message with no registered callback.
    Sending negative response.", self.__log_prefix__())
485                     status = STATUS_BUFFERING_UNHANDLED_REQUEST
486                     data = None
487                 else:
488                     self.logger.debug("%s Executing callback %s to process received data", self.
    __log_prefix__(), callback.__name__)
489                     try:
490                         status, data = callback(msg, *args, **kwargs)
491                     except Exception as e:
492                         self.logger.error('{lp} Exception raised. Check callback {callback_name}:
     "{message}" and it\'s return values for {msg_info}'.format(
493                             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())))
494                         status = STATUS_CALLBACK_ERROR
495                         data = None
496             else:
497                 #
498                 # RESPONSE RECEIVED
499                 #
500                 if callback is None:
501                     self.__buffer_received_data__(msg)
502                 else:
503                     self.logger.debug("%s Executing callback %s to process received data", self.
    __log_prefix__(), callback.__name__)
504                     try:
505                         callback(msg, *args, **kwargs)
506                     except Exception as e:
507                         self.logger.error('{lp} Exception raised. Check callback {callback_name}:
     "{message}" for {msg_info}'.format(lp=self.__log_prefix__(
508                         ), callback_name=callback.__name__, message=str(e), msg_info=self.
    __get_message_name__(msg.get_service_id(), msg.get_data_id())))
509                 return  # No response needed
510         self.send(self.__sid_response_dict__[msg.get_service_id()], msg.get_data_id(), data,
    status=status)
511
512     def __get_message_name__(self, service_id, data_id):
513         return 'service: %s, data_id: %s' % (
514             self.__sid_name_dict__.get(service_id, repr(service_id)),
515             self.__did_name_dict__.get(service_id, {}).get(data_id, repr(data_id)),
516         )
517
518     def __get_status_name__(self, status):
519         return 'status: %s' % (self.__status_name_dict.get(status, 'unknown status: %s' % repr(
    status)))
520
521     def __init_channel_name__(self, channel_name):
522         self.__comm_inst__.init_channel_name(channel_name)
523         self.__callbacks__.init_channel_name(channel_name)
524         if channel_name is None:
525             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + self.
    DEFAULT_CHANNEL_NAME)
```

```
526         else:
527             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' +
    channel_name)
528 
529     def __log_prefix__(self):
530         return 'prot-client:' if self.__comm_inst__.IS_CLIENT else 'prot-server:'
531 
532     def __mk_msg__(self, status, service_id, data_id, data):
533         return data_storage({data_storage.KEY_DATA_ID: data_id, data_storage.KEY_SERVICE_ID:
    service_id, data_storage.KEY_STATUS: status, data_storage.KEY_DATA: data})
534 
535     def __status_log_lvl__(self, status):
536         return STATUS_LOG_LVL.get(status, logging.CRITICAL)
537 
538     def add_data(self, service_id, data_id, name):
539         """
540         Method to add a name for a specific message.
541 
542         :param service_id: The Service-ID of the message. See class definitions starting with ``
    SID_``.
543         :type service_id: int or list of ints
544         :param data_id: The Data-ID of the message.
545         :type data_id: int
546         :param name: The Name for the transfered message.
547         :type name: str
548         """
549         try:
550             iter(service_id)
551         except Exception:
552             service_id = (service_id, )
553 
554         for sid in service_id:
555             if sid not in self.__did_name_dict__:
556                 self.__did_name_dict__[sid] = {}
557             self.__did_name_dict__[sid][data_id] = name
558 
559     def add_msg_to_auth_whitelist_(self, service_id, data_id):
560         """
561         Method to add a specific message to the list, where no authentification is required.
562 
563         :param service_id: The Service-ID of the message. See class definitions starting with ``
    SID_``.
564         :type service_id: int
565         :param data_id: The Data-ID of the message.
566         :type data_id: int
567         """
568         if service_id not in self.__auth_whitelist__:
569             self.__auth_whitelist__[service_id] = []
570         self.__auth_whitelist__[service_id].append(data_id)
571         self.logger.debug('%s Adding Message (%s) to the authentification whitelist',
572                           self.__log_prefix__(), self.__get_message_name__(service_id, data_id))
573 
574     def add_service(self, req_sid, resp_sid, req_name=None, resp_name=None):
575         """
576         Method to add a Service defined by Request- and Response Serivce-ID.
577 
578         :param req_sid: The Request Service-ID.
579         :type req_sid: int
580         :param resp_sid: The Response Service-ID.
581         :type resp_sid: int
582         """
```

```
583         if req_sid in self.__sid_response_dict__:
584             self.logger.error('%s Service with Request-SID=%d and Response-SID=%d not added,
    because request SID is already registered',
585                               self.__log_prefix__(), req_sid, resp_sid)
586             raise RequestSidExistsError("Request for this Service is already registered")
587         elif resp_sid in self.__sid_response_dict__.values():
588             self.logger.error('%s Service with Request-SID=%d and Response-SID=%d not added,
    because response SID is already registered',
589                               self.__log_prefix__(), req_sid, resp_sid)
590             raise ResponseSidExistsError("Response for this Service is already registered")
591         else:
592             self.__sid_response_dict__[req_sid] = resp_sid
593             if req_name is not None:
594                 self.__sid_name_dict__[req_sid] = req_name
595             if resp_name is not None:
596                 self.__sid_name_dict__[resp_sid] = resp_name
597             self.logger.debug('%s Adding Service with Request=%s and Response=%s', self.
    __log_prefix__(),
598                               req_name or repr(req_sid), resp_name or repr(resp_sid))
599 
600     def add_status(self, status, name):
601         """
602         Method to add a name for a status.
603 
604         :param status: The Status. See class definitions starting with ``STATUS_``.
605         :type status: int
606         :param name: The Name for the Status.
607         :type name: str
608         """
609         self.__status_name_dict[status] = name
610 
611     def authentificate(self, timeout=2):
612         """
613         This method authetificates the client at the server.
614 
615         :param timeout: The timeout for the authentification (requesting seed, sending key and
    getting authentification_feedback).
616         :type timeout: float
617         :returns: True, if authentification was successfull; False, if not.
618         :rtype: bool
619 
620         .. note:: An authentification will only processed, if a secret had been given on
    initialisation.
621 
622         .. note:: Client and Server needs to use the same secret.
623         """
624         if self.__secret__ is not None:
625             self.__authentification_state__ = AUTH_STATE_SEED_REQUESTED
626             self.send(SID_AUTH_REQUEST, DID_AUTH_SEED, None)
627             cnt = 0
628             while cnt < timeout * 10:
629                 time.sleep(0.1)
630                 if self.__authentification_state__ == AUTH_STATE_TRUSTED_CONNECTION:
631                     return True
632                 elif self.__authentification_state__ == AUTH_STATE_UNTRUSTED_CONNECTION:
633                     break
634                 cnt += 1
635         return False
636 
637     def check_authentification_state(self):
```

```
638         """
639         This Method return the Authitification State as boolean value.
640 
641         :return: True, if authentification state is okay, otherwise False
642         :rtype: bool
643         """
644         return self.__secret__ is None or self.__authentification_state__ ==
    AUTH_STATE_TRUSTED_CONNECTION
645 
646     def connection_established(self):
647         """
648         This Method returns the Connection state including authentification as a boolean value.
649 
650         :return: True, if the connection is established (incl. authentification, if a secret has
    been given)
651         :rtype: bool
652         """
653         return self.is_connected() and (self.__secret__ is None or self.
    check_authentification_state())
654 
655     def is_connected(self):
656         """
657         This Methods returns Connection state of the Communication Instance :func:`comm_instance.
    is_connected`.
658 
659         :return: True if the :class:`comm_instance` is connected, otherwise False..
660         :rtype: bool
661         """
662         return self.__comm_inst__.is_connected()
663 
664     def receive(self, service_id, data_id, timeout=1):
665         """
666         This Method returns a message object for a defined message or None, if this message is
    not available after the given timout.
667 
668         :param service_id: The Service-ID for the message. See class definitions starting with ``
    SID_``.
669         :type service_id: int
670         :param data_id: The Data-ID for the message.
671         :type data_id: int
672         :param timeout: The timeout for receiving.
673         :type timeout: float
674         :returns: The received data storage object or None, if no data was received.
675         :rtype: data_storage
676         """
677         data = None
678         cnt = 0
679         while data is None and cnt < timeout * 10:
680             try:
681                 data = self.__msg_buffer__.get(service_id, {}).get(data_id, []).pop(0)
682             except IndexError:
683                 data = None
684             cnt += 1
685             time.sleep(0.1)
686         if data is None and cnt >= timeout * 10:
687             self.logger.warning('%s TIMEOUT (%ss): Requested data (service_id: %s; data_id: %s)
    not in buffer.',
688                                 self.__log_prefix__(), repr(timeout), repr(service_id), repr(
    data_id))
689         return data
690 
691     def reconnect(self):
```

```
692         """
693         This methods initiates a reconnect by calling :func:`comm_instance.reconnect`.
694         """
695         return self.__comm_inst__.reconnect()
696 
697     def register_callback(self, service_id, data_id, callback, *args, **kwargs):
698         """
699         This method registers a callback for the given parameters. Giving ``None`` means, that all Service-IDs or all Data-IDs are used.
700         If a message hitting these parameters has been received, the callback will be executed.
701 
702         :param service_id: The Service-ID for the message. See class definitions starting with ``SID_``.
703         :type service_id: int
704         :param data_id: The Data-ID for the message.
705         :type data_id: int
706 
707         .. note:: The :func:`callback` is priorised in the following order:
708 
709             * Callbacks with defined Service-ID and Data-ID.
710             * Callbacks with a defined Service-ID and all Data-IDs.
711             * Callbacks with a defined Data-ID and all Service-IDs.
712             * Unspecific Callbacks.
713 
714         .. note:: The :func:`callback` is executed with these arguments:
715 
716             **Parameters given at the callback call:**
717 
718             * The first Arguments is the received message as :class:`data_storage` object.
719             * Further arguments given at registration.
720             * Further keyword arguments given at registration.
721 
722             **Return value of the callback:**
723 
724             If the Callback is a Request Callback for a registered Service, the return value has to be a tuple or list with
725 
726             * :const:`response_status`: The response status (see class definitions starting with :const:`STA_*`.
727             * :const:`response_data`: A JSON iterable object to be used as data for the response.
728 
729         .. note:: Only registered services will respond via the callbacks return values with the same data_id.
730         """
731         self.__callbacks__.add(service_id, data_id, callback, *args, **kwargs)
732 
733     def send(self, service_id, data_id, data, status=STATUS_OKAY, timeout=2):
734         """
735         This methods sends out a message with the given content.
736 
737         :param service_id: The Service-ID for the message. See class definitions starting with ``SERVICE_``.
738         :type service_id: int
739         :param data_id: The Data-ID for the message.
740         :type data_id: int
741         :param data: The data to be transfered. The data needs to be json compatible.
742         :type data: str
743         :param status: The Status for the message. All requests should have ``STATUS_OKAY``.
744         :type status: int
745         :param timeout: The timeout for sending data (e.g. time to establish new connection).
746         :type timeout: float
```

```
747         :return: True if data had been sent, otherwise False.
748         :rtype: bool
749         """
750         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):
751             msg = data_storage(service_id=service_id, data_id=data_id, data=data, status=status)
752             self.__log_msg__(msg, 'TX ->')
753             return self.__comm_inst__.send(self.__build_frame__(msg), timeout=timeout)
754         else:
755             # Authentification required
756             self.logger.warning("%s Authentification is required. TX-Message %s, %s, data: %s
    will be ignored.", self.__log_prefix__(),
757                                 self.__get_message_name__(service_id, data_id), self.
    __get_status_name__(status), repr(data))
758             return False
759
760
761 class struct_json_protocol(pure_json_protocol):
762     """
763     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.
764     See also parent :py:class:`pure_json_protocol`.
765
766     .. note::
767         This class is depreceated and here for compatibility reasons (to support old clients or
    servers). Usage of :class:`pure_json_protocol` is recommended.
768     """
769
770     def __init__(self, *args, **kwargs):
771         pure_json_protocol.__init__(self, *args, **kwargs)
772
773     def __analyse_frame__(self, frame):
774         status, service_id, data_id = struct.unpack('>III', frame[0:12])
775         data = json.loads(frame[12:-1].decode('utf-8'))
776         return self.__mk_msg__(status, service_id, data_id, data)
777
778     def __build_frame__(self, msg):
779         frame = struct.pack('>III', msg.get_status(), msg.get_service_id(), msg.get_data_id())
780         frame += bytes(json.dumps(msg.get_data()), 'utf-8')
781         frame += self.__calc_chksum__(frame)
782         return frame
783
784     def __calc_chksum__(self, raw_data):
785         chksum = 0
786         for b in raw_data:
787             chksum ^= b
788         return bytes([chksum])
789
790     def __check_frame_checksum__(self, frame):
791         return self.__calc_chksum__(frame[:-1]) == frame[-1:]
```