Unittest for `socket_protocol`

December 21, 2020

Contents

1 Test Information

1.1 Test Candidate Information

The Module socket_protocol is designed to pack and unpack data for serial transportation. For more Information read the sphinx documentation.

Library Information	
Name	socket_protocol
State	Released
Supported Interpreters	python2, python3
Version	b6a37b5ebade9bbfaf8d4b1ce203f822
Dependencies	
stringtools	3eac28a80770a728e1f521fadb92868d

1.2 Unittest Information

Unittest Information	
Version	cd82b7d4eb571a53181f2cb9c7b37417
Testruns with	python 2.7.18 (final), python 3.8.5 (final)

1.3 Test System Information

System Information	
Architecture	64bit
Distribution	Linux Mint 20 ulyana
Hostname	ahorn
Kernel	5.4.0-58-generic (#64-Ubuntu SMP Wed Dec 9 08:16:25 UTC 2020)
Machine	x86_64
Path	/user_data/data/dirk/prj/unittest/socket_protocol/unittest
System	Linux
Username	dirk

2 Statistic

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

Number of tests	**15**
Number of successfull tests	**15**
Number of possibly failed tests	**0**
Number of failed tests	**0**
Executionlevel	Full Test (all defined tests)
Time consumption	11.324s

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

Number of tests	**15**
Number of successfull tests	**15**
Number of possibly failed tests	**0**
Number of failed tests	**0**
Executionlevel	Full Test (all defined tests)
Time consumption	11.332s

2.3 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
socket_protocol	97.8%	96.8%
socket_protocol.__init__.py	97.8%	

3 Testcases with no corresponding Requirement

3.1 Summary for testrun with python 2.7.18 (final)

3.1.1 socket_protocol.pure_json_protocol: Authentification processed without secret.

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (44)
Start-Time:	2020-12-21 22:33:00,256
Finished-Time:	2020-12-21 22:33:00,257
Time-Consumption	0.001s

Testsummary:

Info	Authentification with no secret definition (pure_json_protocol).
Success	Return value of authentification is correct (Content False and Type is <type 'bool'>).

3.1.2 socket_protocol.pure_json_protocol: Authentification required, but not processed/correctly processed.

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (42)
Start-Time:	2020-12-21 22:32:58,845
Finished-Time:	2020-12-21 22:33:00,255
Time-Consumption	1.409s

Testsummary:

Info	Authentification with different secrets for request and response instance (pure_json_protocol).
Success	Return value of authentification is correct (Content False and Type is <type 'bool'>).
Success	Return value of send method is correct (Content True and Type is <type 'bool'>).
Success	Response Status (Authentification required) transfered via pure_json_protocol is correct (Content 2 and Type is <type 'int'>).
Success	Response Data (no data) transfered via pure_json_protocol is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

3.1.3 socket_protocol.pure_json_protocol: Checksum corumpation while sending.

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (36)
Start-Time:	2020-12-21 22:32:56,220
Finished-Time:	2020-12-21 22:32:56,727
Time-Consumption	0.506s

Testsummary:

Info	Send data with wrong checksum by pure_json_protocol.
Success	Return value of send method is correct (Content True and Type is <type 'bool'>).
Success	Callback executed variable is correct (Content False and Type is <type 'bool'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

3.1.4 socket_protocol.pure_json_protocol: Incompatible Callback return value(s).

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (45)
Start-Time:	2020-12-21 22:33:00,257
Finished-Time:	2020-12-21 22:33:00,666
Time-Consumption	0.409s

Testsummary:

Info	Send and received data with incompatible callback (pure_json_protocol).
Success	Exception (TypeError) detection variable is correct (Content True and Type is <type 'bool'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).
Success	Exception (TypeError) detection variable is correct (Content True and Type is <type 'bool'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

3.1.5 socket_protocol.pure_json_protocol: No Callback at response instance for the request.

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (41)
Start-Time:	2020-12-21 22:32:58,136

Finished-Time:	2020-12-21 22:32:58,845
Time-Consumption	0.709s

Testsummary:

Info	Send data, but no callback registered (pure_json_protocol).
Success	Return value of send method is correct (Content True and Type is <type 'bool'>).
Success	Response Status (Request has no callback. Data buffered.) transfered via pure_json_protocol is correct (Content 1 and Type is <type 'int'>).
Success	Response Data (no data) transfered via pure_json_protocol is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

3.1.6 socket_protocol.pure_json_protocol: Register a Callback which is already defined.

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (43)
Start-Time:	2020-12-21 22:33:00,255
Finished-Time:	2020-12-21 22:33:00,256
Time-Consumption	0.001s

Testsummary:

Info	Registering two callbacks which overlap for at least one message (pure_json_protocol).
Success	Exception (RegistrationError) detection variable is correct (Content True and Type is <type 'bool'>).

3.1.7 socket_protocol.pure_json_protocol: Register a second Callback with the same service_id.

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (31)
Start-Time:	2020-12-21 22:32:54,298
Finished-Time:	2020-12-21 22:32:55,006
Time-Consumption	0.709s

Testsummary:

Info	Send and received data by pure_json_protocol.
Success	Return value of send method is correct (Content True and Type is <type 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

3.1.8 socket_protocol.pure_json_protocol: Send and receive check including authentification.

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (27)
Start-Time:	2020-12-21 22:32:50,753
Finished-Time:	2020-12-21 22:32:52,168
Time-Consumption	1.414s

Testsummary:

Info	Send and received data by pure_json_protocol.
Success	Return value of authentification is correct (Content True and Type is <type 'bool'>).
Success	Return value of send method is correct (Content True and Type is <type 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).
Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

3.1.9 socket_protocol.pure_json_protocol: Send and receive check.

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (26)
Start-Time:	2020-12-21 22:32:50,045
Finished-Time:	2020-12-21 22:32:50,753
Time-Consumption	0.707s

Testsummary:

Info	Send and received data by pure_json_protocol.
Success	Return value of send method is correct (Content True and Type is <type 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).
Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

3.1.10 socket_protocol.pure_json_protocol: Timeout measurement for authentification and send method.

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (37)
Start-Time:	2020-12-21 22:32:56,727
Finished-Time:	2020-12-21 22:32:58,136
Time-Consumption	1.409s

Testsummary:

Success	Timeout for authentification is correct (Content 0.20090699195861816 in [0.2 ... 0.22000000000000003] and Type is <type 'float'>).
Success	Timeout for authentification is correct (Content 0.5018911361694336 in [0.5 ... 0.55] and Type is <type 'float'>).
Success	Timeout for send method is correct (Content 0.20077204704284668 in [0.2 ... 0.22000000000000003] and Type is <type 'float'>).
Success	Timeout for send method is correct (Content 0.5015850067138672 in [0.5 ... 0.55] and Type is <type 'float'>).

3.1.11 socket_protocol.pure_json_protocol: Wildcard Callback registration for data_id.

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (30)
Start-Time:	2020-12-21 22:32:53,587
Finished-Time:	2020-12-21 22:32:54,297
Time-Consumption	0.711s

Testsummary:

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for .

Success	Return value of send method is correct (Content True and Type is <type 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).
Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

3.1.12 socket_protocol.pure_json_protocol: Wildcard Callback registration for service_id and data_id.

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (28)
Start-Time:	2020-12-21 22:32:52,168
Finished-Time:	2020-12-21 22:32:52,876
Time-Consumption	0.708s

Testsummary:

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for service_id and data_id.
Success	Return value of send method is correct (Content True and Type is <type 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).
Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

3.1.13 socket_protocol.pure_json_protocol: Wildcard Callback registration for service_id.

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (29)

Start-Time:	2020-12-21 22:32:52,876
Finished-Time:	2020-12-21 22:32:53,586
Time-Consumption	0.710s

Testsummary:

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for service_id.
Success	Return value of send method is correct (Content True and Type is <type 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).
Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

3.1.14 socket_protocol.struct_json_protocol: Send and receive check (Twice for coverage of buffer initialisation).

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (32)
Start-Time:	2020-12-21 22:32:55,007
Finished-Time:	2020-12-21 22:32:56,220
Time-Consumption	1.213s

Testsummary:

Info	Send and received data by struct_json_protocol.
Success	Return value of send method is correct (Content True and Type is <type 'bool'>).
Success	Request Status (Okay) transfered via struct_json_protocol is correct (Content 0 and Type is <type 'int'>).
Success	Request Data transfered via struct_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).
Success	Response Status (Operation not permitted) transfered via struct_json_protocol is correct (Content 5 and Type is <type 'int'>).
Success	Response Data transfered via struct_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).
Success	Return Value (request instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

3.1.15 socket_protocol.struct_json_protocol: Send and receive check.

Testresult

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

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (25)
Start-Time:	2020-12-21 22:32:49,336
Finished-Time:	2020-12-21 22:32:50,045
Time-Consumption	0.709s

Testsummary:

Info	Send and received data by struct_json_protocol.
Success	Return value of send method is correct (Content True and Type is <type 'bool'>).
Success	Request Status (Okay) transfered via struct_json_protocol is correct (Content 0 and Type is <type 'int'>).
Success	Request Data transfered via struct_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).
Success	Response Status (Operation not permitted) transfered via struct_json_protocol is correct (Content 5 and Type is <type 'int'>).
Success	Response Data transfered via struct_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).
Success	Return Value (request instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).
Success	Return Value (response instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

3.2 Summary for testrun with python 3.8.5 (final)

3.2.1 socket_protocol.pure_json_protocol: Authentification processed without secret.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (44)
Start-Time:	2020-12-21 22:33:12,089
Finished-Time:	2020-12-21 22:33:12,091
Time-Consumption	0.002s

Testsummary:

Info	Authentification with no secret definition (pure_json_protocol).
Success	Return value of authentification is correct (Content False and Type is <class 'bool'>).

3.2.2 socket_protocol.pure_json_protocol: Authentification required, but not processed/correctly processed.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (42)
Start-Time:	2020-12-21 22:33:10,675
Finished-Time:	2020-12-21 22:33:12,087
Time-Consumption	1.412s

Testsummary:

Info	Authentification with different secrets for request and response instance (pure_json_protocol).
Success	Return value of authentification is correct (Content False and Type is <class 'bool'>).
Success	Return value of send method is correct (Content True and Type is <class 'bool'>).
Success	Response Status (Authentification required) transfered via pure_json_protocol is correct (Content 2 and Type is <class 'int'>).
Success	Response Data (no data) transfered via pure_json_protocol is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

3.2.3 socket_protocol.pure_json_protocol: Checksum corumpation while sending.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (36)
Start-Time:	2020-12-21 22:33:08,047
Finished-Time:	2020-12-21 22:33:08,557
Time-Consumption	0.510s

Testsummary:

Info	Send data with wrong checksum by pure_json_protocol.
Success	Return value of send method is correct (Content True and Type is <class 'bool'>).
Success	Callback executed variable is correct (Content False and Type is <class 'bool'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

3.2.4 socket_protocol.pure_json_protocol: Incompatible Callback return value(s).

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (45)
Start-Time:	2020-12-21 22:33:12,091

Finished-Time:	2020-12-21 22:33:12,502
Time-Consumption	0.410s

Testsummary:

Info	Send and received data with incompatible callback (pure_json_protocol).
Success	Exception (TypeError) detection variable is correct (Content True and Type is <class 'bool'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).
Success	Exception (TypeError) detection variable is correct (Content True and Type is <class 'bool'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

3.2.5 socket_protocol.pure_json_protocol: No Callback at response instance for the request.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (41)
Start-Time:	2020-12-21 22:33:09,968
Finished-Time:	2020-12-21 22:33:10,674
Time-Consumption	0.706s

Testsummary:

Info	Send data, but no callback registered (pure_json_protocol).
Success	Return value of send method is correct (Content True and Type is <class 'bool'>).
Success	Response Status (Request has no callback. Data buffered.) transfered via pure_json_protocol is correct (Content 1 and Type is <class 'int'>).
Success	Response Data (no data) transfered via pure_json_protocol is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

3.2.6 socket_protocol.pure_json_protocol: Register a Callback which is already defined.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (43)
Start-Time:	2020-12-21 22:33:12,087
Finished-Time:	2020-12-21 22:33:12,089

Time-Consumption	0.001s

Testsummary:

Info	Registering two callbacks which overlap for at least one message (pure_json_protocol).
Success	Exception (RegistrationError) detection variable is correct (Content True and Type is <class 'bool'>).

3.2.7 socket_protocol.pure_json_protocol: Register a second Callback with the same service_id.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (31)
Start-Time:	2020-12-21 22:33:06,124
Finished-Time:	2020-12-21 22:33:06,834
Time-Consumption	0.710s

Testsummary:

Info	Send and received data by pure_json_protocol.
Success	Return value of send method is correct (Content True and Type is <class 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).
Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

3.2.8 socket_protocol.pure_json_protocol: Send and receive check including authentification.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (27)
Start-Time:	2020-12-21 22:33:02,581
Finished-Time:	2020-12-21 22:33:03,994
Time-Consumption	1.413s

Testsummary:

Info	Send and received data by pure_json_protocol.
Success	Return value of authentification is correct (Content True and Type is <class 'bool'>).

Success	Return value of send method is correct (Content True and Type is <class 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).
Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

3.2.9 socket_protocol.pure_json_protocol: Send and receive check.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (26)
Start-Time:	2020-12-21 22:33:01,871
Finished-Time:	2020-12-21 22:33:02,580
Time-Consumption	0.709s

Testsummary:

Info	Send and received data by pure_json_protocol.
Success	Return value of send method is correct (Content True and Type is <class 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).
Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

3.2.10 socket_protocol.pure_json_protocol: Timeout measurement for authentification and send method.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (37)

Start-Time:	2020-12-21 22:33:08,558
Finished-Time:	2020-12-21 22:33:09,968
Time-Consumption	1.410s

Testsummary:

Success	Timeout for authentification is correct (Content 0.20124530792236328 in [0.2 ... 0.22000000000000003] and Type is <class 'float'>).
Success	Timeout for authentification is correct (Content 0.5021066665649414 in [0.5 ... 0.55] and Type is <class 'float'>).
Success	Timeout for send method is correct (Content 0.20095443725585938 in [0.2 ... 0.22000000000000003] and Type is <class 'float'>).
Success	Timeout for send method is correct (Content 0.5017862319946289 in [0.5 ... 0.55] and Type is <class 'float'>).

3.2.11 socket_protocol.pure_json_protocol: Wildcard Callback registration for data_id.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (30)
Start-Time:	2020-12-21 22:33:05,415
Finished-Time:	2020-12-21 22:33:06,124
Time-Consumption	0.709s

Testsummary:

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for .
Success	Return value of send method is correct (Content True and Type is <class 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).
Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

3.2.12 socket_protocol.pure_json_protocol: Wildcard Callback registration for service_id and data_id.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (28)
Start-Time:	2020-12-21 22:33:03,995

Finished-Time:	2020-12-21 22:33:04,704
Time-Consumption	0.709s

Testsummary:

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for service_id and data_id.
Success	Return value of send method is correct (Content True and Type is <class 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).
Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

3.2.13 socket_protocol.pure_json_protocol: Wildcard Callback registration for service_id.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (29)
Start-Time:	2020-12-21 22:33:04,705
Finished-Time:	2020-12-21 22:33:05,414
Time-Consumption	0.709s

Testsummary:

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for service_id.
Success	Return value of send method is correct (Content True and Type is <class 'bool'>).
Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).
Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).
Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).
Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).
Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

3.2.14 socket_protocol.struct_json_protocol: Send and receive check (Twice for coverage of buffer initialisation).

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (32)
Start-Time:	2020-12-21 22:33:06,835
Finished-Time:	2020-12-21 22:33:08,047
Time-Consumption	1.212s

Testsummary:

Info	Send and received data by struct_json_protocol.
Success	Return value of send method is correct (Content True and Type is <class 'bool'>).
Success	Request Status (Okay) transfered via struct_json_protocol is correct (Content 0 and Type is <class 'int'>).
Success	Request Data transfered via struct_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).
Success	Response Status (Operation not permitted) transfered via struct_json_protocol is correct (Content 5 and Type is <class 'int'>).
Success	Response Data transfered via struct_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).
Success	Return Value (request instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

3.2.15 socket_protocol.struct_json_protocol: Send and receive check.

Testresult

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

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (25)
Start-Time:	2020-12-21 22:33:01,162
Finished-Time:	2020-12-21 22:33:01,870
Time-Consumption	0.709s

Testsummary:

Info	Send and received data by struct_json_protocol.
Success	Return value of send method is correct (Content True and Type is <class 'bool'>).
Success	Request Status (Okay) transfered via struct_json_protocol is correct (Content 0 and Type is <class 'int'>).
Success	Request Data transfered via struct_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).
Success	Response Status (Operation not permitted) transfered via struct_json_protocol is correct (Content 5 and Type is <class 'int'>).
Success	Response Data transfered via struct_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).

Success	Return Value (request instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).
Success	Return Value (response instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

A Trace for testrun with python 2.7.18 (final)

A.1 Tests with status Info (15)

A.1.1 socket_protocol.struct_json_protocol: Send and receive check.

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

Info	Send and received data by struct_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
Send data: (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
Receive data (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 45054, data: "[1, 3, u's']"
Send data: (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
Receive data (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
SJP: RX <- status: 5, service_id: 11, data_id: 45054, data: "[1, 3, u's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<type 'bool'>)
Expectation (Return value of send method): result = True (<type 'bool'>)
```

Success	Request Status (Okay) transfered via struct_json_protocol is correct (Content 0 and Type is <type 'int'>).

```
Result (Request Status (Okay) transfered via struct_json_protocol): 0 (<type 'int'>)
Expectation (Request Status (Okay) transfered via struct_json_protocol): result = 0 (<type
↪  'int'>)
```

Success	Request Data transfered via struct_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).

```
Result (Request Data transfered via struct_json_protocol): { u'test': u'test' } (<type
↪  'dict'>)
Expectation (Request Data transfered via struct_json_protocol): result = { u'test': u'test' }
↪  (<type 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via struct_json_protocol is correct (Content 5 and Type is <type 'int'>).

```
Result (Response Status (Operation not permitted) transfered via struct_json_protocol): 5
↪  (<type 'int'>)
Expectation (Response Status (Operation not permitted) transfered via struct_json_protocol):
↪  result = 5 (<type 'int'>)
```

Success	Response Data transfered via struct_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).

```
Result (Response Data transfered via struct_json_protocol): [ 1, 3, u's' ] (<type 'list'>)
Expectation (Response Data transfered via struct_json_protocol): result = [ 1, 3, u's' ]
↪  (<type 'list'>)
```

Success	Return Value (request instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for struct_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
Expectation (Return Value (request instance) for struct_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
Result (Return Value (response instance) for struct_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
Expectation (Return Value (response instance) for struct_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<type 'NoneType'>)
```

A.1.2 socket_protocol.pure_json_protocol: Send and receive check.

Testresult

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

Info	Send and received data by pure_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
Send data: (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 63 ee c4 24
Receive data (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 63 ee c4 24
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 45054, data: "[1, 3, u's']"
Send data: (74): 7b 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 2c 20 22 64
↪  61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 85 0b b7 34
Receive data (74): 7b 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 2c 20 22 64
↪  61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 85 0b b7 34
SJP: RX <- status: 5, service_id: 11, data_id: 45054, data: "[1, 3, u's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<type 'bool'>)
Expectation (Return value of send method): result = True (<type 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<type 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<type
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { u'test': u'test' } (<type 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { u'test': u'test' }
↪  (<type 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<type 'int'>)
```

```
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<type 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, u's' ] (<type 'list'>)
```

```
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, u's' ] (<type
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
```

```
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
```

```
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
```

```
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
```

```
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<type 'NoneType'>)
```

A.1.3 socket_protocol.pure_json_protocol: Send and receive check including authentification.

Testresult

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

Info	Send and received data by pure_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Requesting seed for authentification
SJP: TX -> status: 0, service_id: 1, data_id: 0, data: "None"
Send data: (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22 3a
↪  20 30 7d 2c 2d 2e 5d
Receive data (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22 3a
↪  20 30 7d 2c 2d 2e 5d
SJP: RX <- status: 0, service_id: 1, data_id: 0, data: "None"
SJP: Executing callback __authentificate_create_seed__ to process received data
SJP: Got seed request, sending seed for authentification
SJP: TX -> status: 0, service_id: 2, data_id: 0, data:
↪  "'0922d9bd6ece45f73011f7db3c5da2acbb6167bdb3e7b2dcecad4e98f15adf13'"
Send data: (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 32 2c 20 22 64 61 74 61 22 3a 20 22 30 39 32 32 64 39 62 64 36 65 63 65 34 35 66
↪  37 33 30 31 31 66 37 64 62 33 63 35 64 61 32 61 63 62 62 36 31 36 37 62 64 62 33 65 37 62
↪  32 64 63 65 63 61 64 34 65 39 38 66 31 35 61 64 66 31 33 22 2c 20 22 64 61 74 61 5f 69 64
↪  22 3a 20 30 7d df d1 ae 17
Receive data (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 32 2c 20 22 64 61 74 61 22 3a 20 22 30 39 32 32 64 39 62 64 36 65 63 65 34 35
↪  66 37 33 30 31 31 66 37 64 62 33 63 35 64 61 32 61 63 62 62 36 31 36 37 62 64 62 33 65 37
↪  62 32 64 63 65 63 61 64 34 65 39 38 66 31 35 61 64 66 31 33 22 2c 20 22 64 61 74 61 5f 69
↪  64 22 3a 20 30 7d df d1 ae 17
SJP: RX <- status: 0, service_id: 2, data_id: 0, data:
↪  "u'0922d9bd6ece45f73011f7db3c5da2acbb6167bdb3e7b2dcecad4e98f15adf13'"
SJP: Executing callback __authentificate_create_key__ to process received data
SJP: Got seed, sending key for authentification
SJP: TX -> status: 0, service_id: 3, data_id: 0, data:
↪  "'1464476a2e6a06841bd4922c84ba2f218850ac86c82ae81a81ab132a423137da9ba239acb47cc257740ada0
↪  3035e65a71106341eeb6e7277c1c67846f5b63bf6'"
Send data: (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 33 2c 20 22 64 61 74 61 22 3a 20 22 31 34 36 34 34 37 36 61 32 65 36 61 30 36 38
↪  34 31 62 64 34 39 32 32 63 38 34 62 61 32 66 32 31 38 38 35 30 61 63 38 36 63 38 32 61 65
↪  38 31 61 38 31 61 62 31 33 32 61 34 32 33 31 33 37 64 61 39 62 61 32 33 39 61 63 62 34 37
↪  63 63 32 35 37 37 34 30 61 64 61 30 33 30 33 35 65 36 35 61 37 31 31 30 36 33 34 31 65 65
↪  62 36 65 37 32 37 37 63 31 63 36 37 38 34 36 66 35 62 36 33 62 66 36 22 2c 20 22 64 61 74
↪  61 5f 69 64 22 3a 20 30 7d f6 4d 06 2a
Receive data (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 33 2c 20 22 64 61 74 61 22 3a 20 22 31 34 36 34 34 37 36 61 32 65 36 61 30 36
↪  38 34 31 62 64 34 39 32 32 63 38 34 62 61 32 66 32 31 38 38 35 30 61 63 38 36 63 38 32 61
↪  65 38 31 61 38 31 61 62 31 33 32 61 34 32 33 31 33 37 64 61 39 62 61 32 33 39 61 63 62 34
↪  37 63 63 32 35 37 37 34 30 61 64 61 30 33 30 33 35 65 36 35 61 37 31 31 30 36 33 34 31 65
↪  65 62 36 65 37 32 37 37 63 31 63 36 37 38 34 36 66 35 62 36 33 62 66 36 22 2c 20 22 64 61
↪  74 61 5f 69 64 22 3a 20 30 7d f6 4d 06 2a
SJP: RX <- status: 0, service_id: 3, data_id: 0, data:
↪  "u'1464476a2e6a06841bd4922c84ba2f218850ac86c82ae81a81ab132a423137da9ba239acb47cc257740ada
↪  03035e65a71106341eeb6e7277c1c67846f5b63bf6'"
SJP: Executing callback __authentificate_check_key__ to process received data
```

```
Result (Return value of authentification): True (<type 'bool'>)
Expectation (Return value of authentification): result = True (<type 'bool'>)
```

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

```
Result (Return value of send method): True (<type 'bool'>)
Expectation (Return value of send method): result = True (<type 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<type 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<type
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { u'test': u'test' } (<type 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { u'test': u'test' }
↪  (<type 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<type 'int'>)
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<type 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, u's' ] (<type 'list'>)
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, u's' ] (<type
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
```

```
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
```

```
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<type 'NoneType'>)
```

A.1.4 socket_protocol.pure_json_protocol: Wildcard Callback registration for service_id and data_id.

Testresult

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

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for service_id and data_id.

```
SJP: Cleaning up receive-buffer
```

```
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
```

```
SJP: Cleaning up receive-buffer
```

```
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
```

```
SJP: TX -> status: 0, service_id: 10, data_id: 48879, data: "{u'test': u'test'}"
```

```
Send data: (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d e8 e0 82 59
```

```
Receive data (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d e8 e0 82 59
```

```
SJP: RX <- status: 0, service_id: 10, data_id: 48879, data: "{u'test': u'test'}"
```

```
SJP: Executing callback response_data_method to process received data
```

```
SJP: TX -> status: 5, service_id: 11, data_id: 48879, data: "[1, 3, u's']"
```

```
Send data: (74): 7b 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 2c 20 22 64
↪  61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d 0e 05 f1 49
```

```
Receive data (74): 7b 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 2c 20 22 64
↪  61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d 0e 05 f1 49
```

```
SJP: RX <- status: 5, service_id: 11, data_id: 48879, data: "[1, 3, u's']"
```

```
SJP: Received message has a peculiar status: Operation not permitted
```

```
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<type 'bool'>)
```

```
Expectation (Return value of send method): result = True (<type 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<type 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<type
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { u'test': u'test' } (<type 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { u'test': u'test' }
↪  (<type 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<type 'int'>)
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<type 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, u's' ] (<type 'list'>)
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, u's' ] (<type
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 48879) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<type 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 48879) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<type 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xbeef): result = None (<type 'NoneType'>)
```

A.1.5 socket_protocol.pure_json_protocol: Wildcard Callback registration for service_id.

Testresult

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

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for service_id.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 48879, data: "{u'test': u'test'}"
Send data: (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d e8 e0 82 59
Receive data (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d e8 e0 82 59
SJP: RX <- status: 0, service_id: 10, data_id: 48879, data: "{u'test': u'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 48879, data: "[1, 3, u's']"
Send data: (74): 7b 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 2c 20 22 64
↪  61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d 0e 05 f1 49
Receive data (74): 7b 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 2c 20 22 64
↪  61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d 0e 05 f1 49
SJP: RX <- status: 5, service_id: 11, data_id: 48879, data: "[1, 3, u's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<type 'bool'>)
Expectation (Return value of send method): result = True (<type 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<type 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<type
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { u'test': u'test' } (<type 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { u'test': u'test' }
↪  (<type 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<type 'int'>)
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<type 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, u's' ] (<type 'list'>)
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, u's' ] (<type
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 48879) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<type 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 48879) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<type 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xbeef): result = None (<type 'NoneType'>)
```

A.1.6 socket_protocol.pure_json_protocol: Wildcard Callback registration for data_id.

Testresult

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

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for .

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 48879, data: "{u'test': u'test'}"
Send data: (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d e8 e0 82 59
Receive data (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d e8 e0 82 59
SJP: RX <- status: 0, service_id: 10, data_id: 48879, data: "{u'test': u'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 48879, data: "[1, 3, u's']"
Send data: (74): 7b 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 2c 20 22 64
↪  61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d 0e 05 f1 49
Receive data (74): 7b 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 2c 20 22 64
↪  61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 7d 0e 05 f1 49
SJP: RX <- status: 5, service_id: 11, data_id: 48879, data: "[1, 3, u's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<type 'bool'>)
Expectation (Return value of send method): result = True (<type 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<type 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<type
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { u'test': u'test' } (<type 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { u'test': u'test' }
↪  (<type 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<type 'int'>)
```

```
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<type 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, u's' ] (<type 'list'>)
```

```
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, u's' ] (<type
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 48879) not in buffer.
```

```
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<type 'NoneType'>)
```

```
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 48879) not in buffer.
```

```
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<type 'NoneType'>)
```

```
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xbeef): result = None (<type 'NoneType'>)
```

A.1.7 socket_protocol.pure_json_protocol: Register a second Callback with the same service_id.

Testresult

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

Info	Send and received data by pure_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
Send data: (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 63 ee c4 24
Receive data (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 63 ee c4 24
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
SJP: Executing callback response_data_method_2 to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 45054, data: "[1, 3, u's']"
Send data: (74): 7b 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 2c 20 22 64
↪  61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 85 0b b7 34
Receive data (74): 7b 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 2c 20 22 64
↪  61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 85 0b b7 34
SJP: RX <- status: 5, service_id: 11, data_id: 45054, data: "[1, 3, u's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<type 'bool'>)
Expectation (Return value of send method): result = True (<type 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <type 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<type 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<type
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { u'test': u'test' } (<type 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { u'test': u'test' }
↪  (<type 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <type 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<type 'int'>)
```

```
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<type 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, u's' ] (<type 'list'>)
```

```
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, u's' ] (<type
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
```

```
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
```

```
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
```

```
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
```

```
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<type 'NoneType'>)
```

A.1.8 socket_protocol.struct_json_protocol: Send and receive check (Twice for coverage of buffer initialisation).

Testresult

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

Info	Send and received data by struct_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
Send data: (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
Receive data (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 45054, data: "[1, 3, u's']"
Send data: (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
Receive data (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
SJP: RX <- status: 5, service_id: 11, data_id: 45054, data: "[1, 3, u's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
Send data: (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
Receive data (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 45054, data: "[1, 3, u's']"
Send data: (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
Receive data (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
SJP: RX <- status: 5, service_id: 11, data_id: 45054, data: "[1, 3, u's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<type 'bool'>)
Expectation (Return value of send method): result = True (<type 'bool'>)
```

Success Request Status (Okay) transfered via struct_json_protocol is correct (Content 0 and Type is <type 'int'>).

```
Result (Request Status (Okay) transfered via struct_json_protocol): 0 (<type 'int'>)
Expectation (Request Status (Okay) transfered via struct_json_protocol): result = 0 (<type
↪  'int'>)
```

Success Request Data transfered via struct_json_protocol is correct (Content {u'test': u'test'} and Type is <type 'dict'>).

```
Result (Request Data transfered via struct_json_protocol): { u'test': u'test' } (<type
↪  'dict'>)
```

```
Expectation (Request Data transfered via struct_json_protocol): result = { u'test': u'test' }
↪  (<type 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via struct_json_protocol is correct (Content 5 and Type is <type 'int'>).

```
Result (Response Status (Operation not permitted) transfered via struct_json_protocol): 5
↪  (<type 'int'>)
```

```
Expectation (Response Status (Operation not permitted) transfered via struct_json_protocol):
↪  result = 5 (<type 'int'>)
```

Success	Response Data transfered via struct_json_protocol is correct (Content [1, 3, u's'] and Type is <type 'list'>).

```
Result (Response Data transfered via struct_json_protocol): [ 1, 3, u's' ] (<type 'list'>)
```

```
Expectation (Response Data transfered via struct_json_protocol): result = [ 1, 3, u's' ]
↪  (<type 'list'>)
```

Success	Return Value (request instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
```

```
Result (Return Value (request instance) for struct_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
```

```
Expectation (Return Value (request instance) for struct_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
```

```
Result (Return Value (response instance) for struct_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
```

```
Expectation (Return Value (response instance) for struct_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<type 'NoneType'>)
```

A.1.9 socket_protocol.pure_json_protocol: Checksum corumpation while sending.

Testresult

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

Info	Send data with wrong checksum by pure_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
Send data: (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 63 ee c4 24
Receive data (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 63 ee c4 25
SJP: Received message has a wrong checksum and will be ignored: (79): 7b 22 73 74 61 74 75 73
↪  22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22
↪  3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 2c 20 22 64 61 74 61 5f 69 64 22 3a
↪  20 34 35 30 35 34 7d 63 ee c4 25.
```

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

```
Result (Return value of send method): True (<type 'bool'>)
Expectation (Return value of send method): result = True (<type 'bool'>)
```

Success Callback executed variable is correct (Content False and Type is <type 'bool'>).

```
Result (Callback executed variable): False (<type 'bool'>)
Expectation (Callback executed variable): result = False (<type 'bool'>)
```

Success Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<type 'NoneType'>)
```

Success Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<type 'NoneType'>)
```

A.1.10 socket_protocol.pure_json_protocol: Timeout measurement for authentification and send method.

Testresult

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

Success	Timeout for authentification is correct (Content 0.20090699195861816 in [0.2 ... 0.22000000000000003] and Type is <type 'float'>).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Requesting seed for authentification
SJP: TX -> status: 0, service_id: 1, data_id: 0, data: "None"
Send data: (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22 3a
↪  20 30 7d 2c 2d 2e 5d
Result (Timeout for authentification): 0.20090699195861816 (<type 'float'>)
Expectation (Timeout for authentification): 0.2 <= result <= 0.22000000000000003
```

Success	Timeout for authentification is correct (Content 0.5018911361694336 in [0.5 ... 0.55] and Type is <type 'float'>).

```
SJP: Requesting seed for authentification
SJP: TX -> status: 0, service_id: 1, data_id: 0, data: "None"
Send data: (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22 3a
↪  20 30 7d 2c 2d 2e 5d
Result (Timeout for authentification): 0.5018911361694336 (<type 'float'>)
Expectation (Timeout for authentification): 0.5 <= result <= 0.55
```

Success	Timeout for send method is correct (Content 0.20077204704284668 in [0.2 ... 0.22000000000000003] and Type is <type 'float'>).

```
SJP: TIMEOUT (0.2s): Requested data (service_id: 30; data_id: 0) not in buffer.
Result (Timeout for send method): 0.20077204704284668 (<type 'float'>)
Expectation (Timeout for send method): 0.2 <= result <= 0.22000000000000003
```

Success	Timeout for send method is correct (Content 0.5015850067138672 in [0.5 ... 0.55] and Type is <type 'float'>).

```
SJP: TIMEOUT (0.5s): Requested data (service_id: 30; data_id: 0) not in buffer.
Result (Timeout for send method): 0.5015850067138672 (<type 'float'>)
Expectation (Timeout for send method): 0.5 <= result <= 0.55
```

A.1.11 socket_protocol.pure_json_protocol: No Callback at response instance for the request.

Testresult

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

Info	Send data, but no callback registered (pure_json_protocol).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
Send data: (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 63 ee c4 24
Receive data (79): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22
↪  7d 2c 20 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 7d 63 ee c4 24
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{u'test': u'test'}"
SJP: Received message with no registered callback. Sending negative response.
SJP: TX -> status: 1, service_id: 11, data_id: 45054, data: "None"
Send data: (67): 7b 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22
↪  3a 20 34 35 30 35 34 7d b1 70 10 64
Receive data (67): 7b 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22
↪  3a 20 34 35 30 35 34 7d b1 70 10 64
SJP: RX <- status: 1, service_id: 11, data_id: 45054, data: "None"
SJP: Received message has a peculiar status: Request has no callback. Data buffered.
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<type 'bool'>)
Expectation (Return value of send method): result = True (<type 'bool'>)
```

Success	Response Status (Request has no callback. Data buffered.) transfered via pure_json_protocol is correct (Content 1 and Type is <type 'int'>).

```
Result (Response Status (Request has no callback. Data buffered.) transfered via
↪  pure_json_protocol): 1 (<type 'int'>)
Expectation (Response Status (Request has no callback. Data buffered.) transfered via
↪  pure_json_protocol): result = 1 (<type 'int'>)
```

Success	Response Data (no data) transfered via pure_json_protocol is correct (Content None and Type is <type 'NoneType'>).

```
Result (Response Data (no data) transfered via pure_json_protocol): None (<type 'NoneType'>)
Expectation (Response Data (no data) transfered via pure_json_protocol): result = None (<type
↪  'NoneType'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<type 'NoneType'>)
```

A.1.12 socket_protocol.pure_json_protocol: Authentification required, but not processed/correctly processed.

Testresult

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

Info	Authentification with different secrets for request and response instance (pure_json_protocol).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Requesting seed for authentification
SJP: TX -> status: 0, service_id: 1, data_id: 0, data: "None"
Send data: (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22 3a
↪  20 30 7d 2c 2d 2e 5d
Receive data (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22 3a
↪  20 30 7d 2c 2d 2e 5d
SJP: RX <- status: 0, service_id: 1, data_id: 0, data: "None"
SJP: Executing callback __authentificate_create_seed__ to process received data
SJP: Got seed request, sending seed for authentification
SJP: TX -> status: 0, service_id: 2, data_id: 0, data:
↪  "'0e07fcd8b40cf2c9341c9192b86f6d6f8b014a35fe6a99028dd5014d42f60a36'"
Send data: (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 32 2c 20 22 64 61 74 61 22 3a 20 22 30 65 30 37 66 63 64 38 62 34 30 63 66 32 63
↪  39 33 34 31 63 39 31 39 32 62 38 36 66 36 64 36 66 38 62 30 31 34 61 33 35 66 65 36 61 39
↪  39 30 32 38 64 64 35 30 31 34 64 34 32 66 36 30 61 33 36 22 2c 20 22 64 61 74 61 5f 69 64
↪  22 3a 20 30 7d 0c 1e 5c ae
Receive data (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 32 2c 20 22 64 61 74 61 22 3a 20 22 30 65 30 37 66 63 64 38 62 34 30 63 66 32
↪  63 39 33 34 31 63 39 31 39 32 62 38 36 66 36 64 36 66 38 62 30 31 34 61 33 35 66 65 36 61
↪  39 39 30 32 38 64 64 35 30 31 34 64 34 32 66 36 30 61 33 36 22 2c 20 22 64 61 74 61 5f 69
↪  64 22 3a 20 30 7d 0c 1e 5c ae
SJP: RX <- status: 0, service_id: 2, data_id: 0, data:
↪  "u'0e07fcd8b40cf2c9341c9192b86f6d6f8b014a35fe6a99028dd5014d42f60a36'"
SJP: Executing callback __authentificate_create_key__ to process received data
SJP: Got seed, sending key for authentification
SJP: TX -> status: 0, service_id: 3, data_id: 0, data:
↪  "'8d64836851399cace34c5a414cc6ff2ae92e9dc5bc0f1763e95cf5ace45abc023e0a004554e03650be3cfe8
↪  15f4d4859eeac3d3ae77149a730dff8dc21fa5b8e'"
Send data: (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 33 2c 20 22 64 61 74 61 22 3a 20 22 38 64 36 34 38 33 36 38 35 31 33 39 39 63 61
↪  63 65 33 34 63 35 61 34 31 34 63 63 36 66 66 32 61 65 39 32 65 39 64 63 35 62 63 30 66 31
↪  37 36 33 65 39 35 63 66 35 61 63 65 34 35 61 62 63 30 32 33 65 30 61 30 30 34 35 35 34 65
↪  30 33 36 35 30 62 65 33 63 66 65 38 31 35 66 34 64 34 38 35 39 65 65 61 63 33 64 33 61 65
↪  37 37 31 34 39 61 37 33 30 64 66 66 38 64 63 32 31 66 61 35 62 38 65 22 2c 20 22 64 61 74
↪  61 5f 69 64 22 3a 20 30 7d 66 cf 82 1c
Receive data (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 33 2c 20 22 64 61 74 61 22 3a 20 22 38 64 36 34 38 33 36 38 35 31 33 39 39 63
↪  61 63 65 33 34 63 35 61 34 31 34 63 63 36 66 66 32 61 65 39 32 65 39 64 63 35 62 63 30 66
↪  31 37 36 33 65 39 35 63 66 35 61 63 65 34 35 61 62 63 30 32 33 65 30 61 30 30 34 35 35 34
↪  65 30 33 36 35 30 62 65 33 63 66 65 38 31 35 66 34 64 34 38 35 39 65 65 61 63 33 64 33 61
↪  65 37 37 31 34 39 61 37 33 30 64 66 66 38 64 63 32 31 66 61 35 62 38 65 22 2c 20 22 64 61
↪  74 61 5f 69 64 22 3a 20 30 7d 66 cf 82 1c
SJP: RX <- status: 0, service_id: 3, data_id: 0, data:
↪  "u'8d64836851399cace34c5a414cc6ff2ae92e9dc5bc0f1763e95cf5ace45abc023e0a004554e03650be3cfe
↪  815f4d4859eeac3d3ae77149a730dff8dc21fa5b8e'"
SJP: Executing callback __authentificate_check_key__ to process received data
```

```
Result (Return value of authentification): False (<type 'bool'>)
Expectation (Return value of authentification): result = False (<type 'bool'>)
```

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

```
Result (Return value of send method): True (<type 'bool'>)
Expectation (Return value of send method): result = True (<type 'bool'>)
```

Success	Response Status (Authentification required) transfered via pure_json_protocol is correct (Content 2 and Type is <type 'int'>).

```
Result (Response Status (Authentification required) transfered via pure_json_protocol): 2
↪  (<type 'int'>)
Expectation (Response Status (Authentification required) transfered via pure_json_protocol):
↪  result = 2 (<type 'int'>)
```

Success	Response Data (no data) transfered via pure_json_protocol is correct (Content None and Type is <type 'NoneType'>).

```
Result (Response Data (no data) transfered via pure_json_protocol): None (<type 'NoneType'>)
Expectation (Response Data (no data) transfered via pure_json_protocol): result = None (<type
↪  'NoneType'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<type 'NoneType'>)
```

A.1.13 socket_protocol.pure_json_protocol: Register a Callback which is already defined.

Testresult

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

Info	Registering two callbacks which overlap for at least one message (pure_json_protocol).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
```

Success	Exception (RegistrationError) detection variable is correct (Content True and Type is <type 'bool'>).

```
Result (Exception (RegistrationError) detection variable): True (<type 'bool'>)
Expectation (Exception (RegistrationError) detection variable): result = True (<type 'bool'>)
```

A.1.14 socket_protocol.pure_json_protocol: Authentification processed without secret.

Testresult

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

Info	Authentification with no secret definition (pure_json_protocol).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
```

Success	Return value of authentification is correct (Content False and Type is <type 'bool'>).

```
Result (Return value of authentification): False (<type 'bool'>)
Expectation (Return value of authentification): result = False (<type 'bool'>)
```

A.1.15 socket_protocol.pure_json_protocol: Incompatible Callback return value(s).

Testresult

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

Info	Send and received data with incompatible callback (pure_json_protocol).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "None"
Send data: (67): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22
↪  3a 20 34 35 30 35 34 7d fc 3e bd 5f
Receive data (67): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22
↪  3a 20 34 35 30 35 34 7d fc 3e bd 5f
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "None"
SJP: Executing callback response_data_method_fail to process received data
SJP: TX -> status: 0, service_id: 10, data_id: 48879, data: "None"
Send data: (67): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22
↪  3a 20 34 38 38 37 39 7d 77 30 fb 22
Receive data (67): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22
↪  3a 20 34 38 38 37 39 7d 77 30 fb 22
SJP: RX <- status: 0, service_id: 10, data_id: 48879, data: "None"
SJP: Received message with no registered callback. Sending negative response.
SJP: TX -> status: 1, service_id: 11, data_id: 48879, data: "None"
Send data: (67): 7b 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22
↪  3a 20 34 38 38 37 39 7d 3a 7e 56 19
Receive data (67): 7b 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 73 65 72 76 69 63 65 5f 69 64
↪  22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61 5f 69 64 22
↪  3a 20 34 38 38 37 39 7d 3a 7e 56 19
SJP: RX <- status: 1, service_id: 11, data_id: 48879, data: "None"
SJP: Received message has a peculiar status: Request has no callback. Data buffered.
SJP: Executing callback response_data_method_fail to process received data
```

Success	Exception (TypeError) detection variable is correct (Content True and Type is <type 'bool'>).

```
Result (Exception (TypeError) detection variable): True (<type 'bool'>)
Expectation (Exception (TypeError) detection variable): result = True (<type 'bool'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<type 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<type 'NoneType'>)
```

Success	Exception (TypeError) detection variable is correct (Content True and Type is <type 'bool'>).

```
Result (Exception (TypeError) detection variable): True (<type 'bool'>)
Expectation (Exception (TypeError) detection variable): result = True (<type 'bool'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 48879) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<type 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): result = None (<type 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <type 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 48879) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<type 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xbeef): result = None (<type 'NoneType'>)
```

B Trace for testrun with python 3.8.5 (final)

B.1 Tests with status Info (15)

B.1.1 socket_protocol.struct_json_protocol: Send and receive check.

Testresult

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

Info	Send and received data by struct_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
Send data: (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
Receive data (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 45054, data: "[1, 3, 's']"
Send data: (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
Receive data (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
SJP: RX <- status: 5, service_id: 11, data_id: 45054, data: "[1, 3, 's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<class 'bool'>)
Expectation (Return value of send method): result = True (<class 'bool'>)
```

Success Request Status (Okay) transfered via struct_json_protocol is correct (Content 0 and Type is <class 'int'>).

```
Result (Request Status (Okay) transfered via struct_json_protocol): 0 (<class 'int'>)
Expectation (Request Status (Okay) transfered via struct_json_protocol): result = 0 (<class
↪  'int'>)
```

Success Request Data transfered via struct_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).

```
Result (Request Data transfered via struct_json_protocol): { 'test': 'test' } (<class 'dict'>)
Expectation (Request Data transfered via struct_json_protocol): result = { 'test': 'test' }
↪  (<class 'dict'>)
```

Success Response Status (Operation not permitted) transfered via struct_json_protocol is correct (Content 5 and Type is <class 'int'>).

```
Result (Response Status (Operation not permitted) transfered via struct_json_protocol): 5
↪  (<class 'int'>)
Expectation (Response Status (Operation not permitted) transfered via struct_json_protocol):
↪  result = 5 (<class 'int'>)
```

Success Response Data transfered via struct_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).

```
Result (Response Data transfered via struct_json_protocol): [ 1, 3, 's' ] (<class 'list'>)
```

```
Expectation (Response Data transfered via struct_json_protocol): result = [ 1, 3, 's' ]
↪  (<class 'list'>)
```

Success	Return Value (request instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
```

```
Result (Return Value (request instance) for struct_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
```

```
Expectation (Return Value (request instance) for struct_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
```

```
Result (Return Value (response instance) for struct_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
```

```
Expectation (Return Value (response instance) for struct_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<class 'NoneType'>)
```

B.1.2 socket_protocol.pure_json_protocol: Send and receive check.

Testresult

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

Info	Send and received data by pure_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
Send data: (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d 82 1c 8b 89
Receive data (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d 82 1c 8b 89
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 45054, data: "[1, 3, 's']"
Send data: (74): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 64 61 74 61
↪  22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 7d 60 f8 dc 89
Receive data (74): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 64 61 74 61
↪  22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 7d 60 f8 dc 89
SJP: RX <- status: 5, service_id: 11, data_id: 45054, data: "[1, 3, 's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<class 'bool'>)
Expectation (Return value of send method): result = True (<class 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<class 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<class
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { 'test': 'test' } (<class 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { 'test': 'test' }
↪  (<class 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<class 'int'>)
```

```
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<class 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, 's' ] (<class 'list'>)
```

```
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, 's' ] (<class
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
```

```
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
```

```
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
```

```
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
```

```
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<class 'NoneType'>)
```

B.1.3 socket_protocol.pure_json_protocol: Send and receive check including authentification.

Testresult

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

Info	Send and received data by pure_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Requesting seed for authentification
SJP: TX -> status: 0, service_id: 1, data_id: 0, data: "None"
Send data: (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75
↪  6c 6c 7d 9e 85 7b 8d
Receive data (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75
↪  6c 6c 7d 9e 85 7b 8d
SJP: RX <- status: 0, service_id: 1, data_id: 0, data: "None"
SJP: Executing callback __authentificate_create_seed__ to process received data
SJP: Got seed request, sending seed for authentification
SJP: TX -> status: 0, service_id: 2, data_id: 0, data:
↪  "'efa0dc1516add005e12d4fbebaac0403d96f6e1bb4582d5391a272cf9169bd32'"
Send data: (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 32 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 65
↪  66 61 30 64 63 31 35 31 36 61 64 64 30 30 35 65 31 32 64 34 66 62 65 62 61 61 63 30 34 30
↪  33 64 39 36 66 36 65 31 62 62 34 35 38 32 64 35 33 39 31 61 32 37 32 63 66 39 31 36 39 62
↪  64 33 32 22 7d df 11 5a 26
Receive data (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f
↪  69 64 22 3a 20 32 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22
↪  65 66 61 30 64 63 31 35 31 36 61 64 64 30 30 35 65 31 32 64 34 66 62 65 62 61 61 63 30 34
↪  30 33 64 39 36 66 36 65 31 62 62 34 35 38 32 64 35 33 39 31 61 32 37 32 63 66 39 31 36 39
↪  62 64 33 32 22 7d df 11 5a 26
SJP: RX <- status: 0, service_id: 2, data_id: 0, data:
↪  "'efa0dc1516add005e12d4fbebaac0403d96f6e1bb4582d5391a272cf9169bd32'"
SJP: Executing callback __authentificate_create_key__ to process received data
SJP: Got seed, sending key for authentification
SJP: TX -> status: 0, service_id: 3, data_id: 0, data:
↪  "'8236f10c89bff4f28f00db77091cbc81edaeaf6de1716ce477602e97840605d19f5e67cbfd3378d2cf91635
↪  1fc6f1eac1ff473f6b025a908522a20965e8965b6'"
Send data: (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 33 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 38
↪  32 33 36 66 31 30 63 38 39 62 66 66 34 66 32 38 66 30 30 64 62 37 37 30 39 31 63 62 63 38
↪  31 65 64 61 65 61 66 36 64 65 31 37 31 36 63 65 34 37 37 36 30 32 65 39 37 38 34 30 36 30
↪  35 64 31 39 66 35 65 36 37 63 62 66 64 33 33 37 38 64 32 63 66 39 31 36 33 35 31 66 63 36
↪  66 31 65 61 63 31 66 66 34 37 33 66 36 62 30 32 35 61 39 30 38 35 32 32 61 32 30 39 36 35
↪  65 38 39 36 35 62 36 22 7d a8 8f 70 81
Receive data (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f
↪  69 64 22 3a 20 33 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22
↪  38 32 33 36 66 31 30 63 38 39 62 66 66 34 66 32 38 66 30 30 64 62 37 37 30 39 31 63 62 63
↪  38 31 65 64 61 65 61 66 36 64 65 31 37 31 36 63 65 34 37 37 36 30 32 65 39 37 38 34 30 36
↪  30 35 64 31 39 66 35 65 36 37 63 62 66 64 33 33 37 38 64 32 63 66 39 31 36 33 35 31 66 63
↪  36 66 31 65 61 63 31 66 66 34 37 33 66 36 62 30 32 35 61 39 30 38 35 32 32 61 32 30 39 36
↪  35 65 38 39 36 35 62 36 22 7d a8 8f 70 81
SJP: RX <- status: 0, service_id: 3, data_id: 0, data:
↪  "'8236f10c89bff4f28f00db77091cbc81edaeaf6de1716ce477602e97840605d19f5e67cbfd3378d2cf91635
↪  1fc6f1eac1ff473f6b025a908522a20965e8965b6'"
SJP: Executing callback __authentificate_check_key__ to process received data
```

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

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

```
Result (Return value of send method): True (<class 'bool'>)
Expectation (Return value of send method): result = True (<class 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<class 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<class
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { 'test': 'test' } (<class 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { 'test': 'test' }
↪  (<class 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<class 'int'>)
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<class 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, 's' ] (<class 'list'>)
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, 's' ] (<class
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<class 'NoneType'>)
```

B.1.4 socket_protocol.pure_json_protocol: Wildcard Callback registration for service_id and data_id.

Testresult

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

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for service_id and data_id.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 48879, data: "{'test': 'test'}"
Send data: (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d cc a2 b9 e6
Receive data (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d cc a2 b9 e6
SJP: RX <- status: 0, service_id: 10, data_id: 48879, data: "{'test': 'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 48879, data: "[1, 3, 's']"
Send data: (74): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 64 61 74 61
↪  22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 7d 7d 1e 6e 1b
Receive data (74): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 64 61 74 61
↪  22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 7d 7d 1e 6e 1b
SJP: RX <- status: 5, service_id: 11, data_id: 48879, data: "[1, 3, 's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<class 'bool'>)
Expectation (Return value of send method): result = True (<class 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<class 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<class
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { 'test': 'test' } (<class 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { 'test': 'test' }
↪  (<class 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<class 'int'>)
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<class 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, 's' ] (<class 'list'>)
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, 's' ] (<class
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 48879) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<class 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 48879) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<class 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xbeef): result = None (<class 'NoneType'>)
```

B.1.5 socket_protocol.pure_json_protocol: Wildcard Callback registration for service_id.

Testresult

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

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for service_id.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 48879, data: "{'test': 'test'}"
Send data: (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d cc a2 b9 e6
Receive data (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d cc a2 b9 e6
SJP: RX <- status: 0, service_id: 10, data_id: 48879, data: "{'test': 'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 48879, data: "[1, 3, 's']"
Send data: (74): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 64 61 74 61
↪  22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 7d 7d 1e 6e 1b
Receive data (74): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 64 61 74 61
↪  22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 7d 7d 1e 6e 1b
SJP: RX <- status: 5, service_id: 11, data_id: 48879, data: "[1, 3, 's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<class 'bool'>)
Expectation (Return value of send method): result = True (<class 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<class 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<class
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { 'test': 'test' } (<class 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { 'test': 'test' }
↪  (<class 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<class 'int'>)
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<class 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, 's' ] (<class 'list'>)
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, 's' ] (<class
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 48879) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<class 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 48879) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<class 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xbeef): result = None (<class 'NoneType'>)
```

B.1.6 socket_protocol.pure_json_protocol: Wildcard Callback registration for data_id.

Testresult

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

Info	Send and received data by pure_json_protocol. Wildcard callback registerd for .

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 48879, data: "{'test': 'test'}"
Send data: (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d cc a2 b9 e6
Receive data (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d cc a2 b9 e6
SJP: RX <- status: 0, service_id: 10, data_id: 48879, data: "{'test': 'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 48879, data: "[1, 3, 's']"
Send data: (74): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 64 61 74 61
↪  22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 7d 7d 1e 6e 1b
Receive data (74): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 64 61 74 61
↪  22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 7d 7d 1e 6e 1b
SJP: RX <- status: 5, service_id: 11, data_id: 48879, data: "[1, 3, 's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<class 'bool'>)
Expectation (Return value of send method): result = True (<class 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<class 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<class
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { 'test': 'test' } (<class 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { 'test': 'test' }
↪  (<class 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<class 'int'>)
```

```
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<class 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, 's' ] (<class 'list'>)
```

```
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, 's' ] (<class
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 48879) not in buffer.
```

```
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<class 'NoneType'>)
```

```
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 48879) not in buffer.
```

```
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<class 'NoneType'>)
```

```
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xbeef): result = None (<class 'NoneType'>)
```

B.1.7 socket_protocol.pure_json_protocol: Register a second Callback with the same service_id.

Testresult

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

Info	Send and received data by pure_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
Send data: (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d 82 1c 8b 89
Receive data (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d 82 1c 8b 89
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
SJP: Executing callback response_data_method_2 to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 45054, data: "[1, 3, 's']"
Send data: (74): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 64 61 74 61
↪  22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 7d 60 f8 dc 89
Receive data (74): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 35 2c 20 22 64 61 74 61
↪  22 3a 20 5b 31 2c 20 33 2c 20 22 73 22 5d 7d 60 f8 dc 89
SJP: RX <- status: 5, service_id: 11, data_id: 45054, data: "[1, 3, 's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<class 'bool'>)
Expectation (Return value of send method): result = True (<class 'bool'>)
```

Success	Request Status (Okay) transfered via pure_json_protocol is correct (Content 0 and Type is <class 'int'>).

```
Result (Request Status (Okay) transfered via pure_json_protocol): 0 (<class 'int'>)
Expectation (Request Status (Okay) transfered via pure_json_protocol): result = 0 (<class
↪  'int'>)
```

Success	Request Data transfered via pure_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).

```
Result (Request Data transfered via pure_json_protocol): { 'test': 'test' } (<class 'dict'>)
Expectation (Request Data transfered via pure_json_protocol): result = { 'test': 'test' }
↪  (<class 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via pure_json_protocol is correct (Content 5 and Type is <class 'int'>).

```
Result (Response Status (Operation not permitted) transfered via pure_json_protocol): 5
↪  (<class 'int'>)
```

```
Expectation (Response Status (Operation not permitted) transfered via pure_json_protocol):
↪  result = 5 (<class 'int'>)
```

Success	Response Data transfered via pure_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).

```
Result (Response Data transfered via pure_json_protocol): [ 1, 3, 's' ] (<class 'list'>)
```

```
Expectation (Response Data transfered via pure_json_protocol): result = [ 1, 3, 's' ] (<class
↪  'list'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
```

```
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
```

```
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
```

```
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
```

```
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<class 'NoneType'>)
```

B.1.8 socket_protocol.struct_json_protocol: Send and receive check (Twice for coverage of buffer initialisation).

Testresult

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

Info	Send and received data by struct_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
Send data: (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
Receive data (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 45054, data: "[1, 3, 's']"
Send data: (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
Receive data (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
SJP: RX <- status: 5, service_id: 11, data_id: 45054, data: "[1, 3, 's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
Send data: (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
Receive data (29): 00 00 00 00 00 00 00 0a 00 00 af fe 7b 22 74 65 73 74 22 3a 20 22 74 65 73
↪  74 22 7d 47
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
SJP: Executing callback response_data_method to process received data
SJP: TX -> status: 5, service_id: 11, data_id: 45054, data: "[1, 3, 's']"
Send data: (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
Receive data (24): 00 00 00 05 00 00 00 0b 00 00 af fe 5b 31 2c 20 33 2c 20 22 73 22 5d 28
SJP: RX <- status: 5, service_id: 11, data_id: 45054, data: "[1, 3, 's']"
SJP: Received message has a peculiar status: Operation not permitted
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<class 'bool'>)
Expectation (Return value of send method): result = True (<class 'bool'>)
```

Success Request Status (Okay) transfered via struct_json_protocol is correct (Content 0 and Type is <class 'int'>).

```
Result (Request Status (Okay) transfered via struct_json_protocol): 0 (<class 'int'>)
Expectation (Request Status (Okay) transfered via struct_json_protocol): result = 0 (<class
↪  'int'>)
```

Success Request Data transfered via struct_json_protocol is correct (Content {'test': 'test'} and Type is <class 'dict'>).

```
Result (Request Data transfered via struct_json_protocol): { 'test': 'test' } (<class 'dict'>)
Expectation (Request Data transfered via struct_json_protocol): result = { 'test': 'test' }
↪  (<class 'dict'>)
```

Success	Response Status (Operation not permitted) transfered via struct_json_protocol is correct (Content 5 and Type is <class 'int'>).

```
Result (Response Status (Operation not permitted) transfered via struct_json_protocol): 5
↪  (<class 'int'>)
Expectation (Response Status (Operation not permitted) transfered via struct_json_protocol):
↪  result = 5 (<class 'int'>)
```

Success	Response Data transfered via struct_json_protocol is correct (Content [1, 3, 's'] and Type is <class 'list'>).

```
Result (Response Data transfered via struct_json_protocol): [ 1, 3, 's' ] (<class 'list'>)
Expectation (Response Data transfered via struct_json_protocol): result = [ 1, 3, 's' ]
↪  (<class 'list'>)
```

Success	Return Value (request instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for struct_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (request instance) for struct_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for struct_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
Result (Return Value (response instance) for struct_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (response instance) for struct_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<class 'NoneType'>)
```

B.1.9 socket_protocol.pure_json_protocol: Checksum corumpation while sending.

Testresult

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

Info	Send data with wrong checksum by pure_json_protocol.

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
Send data: (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d 82 1c 8b 89
Receive data (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d 82 1c 8b 8a
SJP: Received message has a wrong checksum and will be ignored: (79): 7b 22 64 61 74 61 5f 69
↪  64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22
↪  73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 7b 22 74 65 73 74 22 3a 20 22
↪  74 65 73 74 22 7d 7d 82 1c 8b 8a.
```

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

```
Result (Return value of send method): True (<class 'bool'>)
Expectation (Return value of send method): result = True (<class 'bool'>)
```

Success Callback executed variable is correct (Content False and Type is <class 'bool'>).

```
Result (Callback executed variable): False (<class 'bool'>)
Expectation (Callback executed variable): result = False (<class 'bool'>)
```

Success Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<class 'NoneType'>)
```

Success Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<class 'NoneType'>)
```

B.1.10 socket_protocol.pure_json_protocol: Timeout measurement for authentification and send method.

Testresult

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

Success	Timeout for authentification is correct (Content 0.20124530792236328 in [0.2 ... 0.22000000000000003] and Type is <class 'float'>).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Requesting seed for authentification
SJP: TX -> status: 0, service_id: 1, data_id: 0, data: "None"
Send data: (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75
↪  6c 6c 7d 9e 85 7b 8d
Result (Timeout for authentification): 0.20124530792236328 (<class 'float'>)
Expectation (Timeout for authentification): 0.2 <= result <= 0.22000000000000003
```

Success	Timeout for authentification is correct (Content 0.5021066665649414 in [0.5 ... 0.55] and Type is <class 'float'>).

```
SJP: Requesting seed for authentification
SJP: TX -> status: 0, service_id: 1, data_id: 0, data: "None"
Send data: (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75
↪  6c 6c 7d 9e 85 7b 8d
Result (Timeout for authentification): 0.5021066665649414 (<class 'float'>)
Expectation (Timeout for authentification): 0.5 <= result <= 0.55
```

Success	Timeout for send method is correct (Content 0.20095443725585938 in [0.2 ... 0.22000000000000003] and Type is <class 'float'>).

```
SJP: TIMEOUT (0.2s): Requested data (service_id: 30; data_id: 0) not in buffer.
Result (Timeout for send method): 0.20095443725585938 (<class 'float'>)
Expectation (Timeout for send method): 0.2 <= result <= 0.22000000000000003
```

Success	Timeout for send method is correct (Content 0.5017862319946289 in [0.5 ... 0.55] and Type is <class 'float'>).

```
SJP: TIMEOUT (0.5s): Requested data (service_id: 30; data_id: 0) not in buffer.
Result (Timeout for send method): 0.5017862319946289 (<class 'float'>)
Expectation (Timeout for send method): 0.5 <= result <= 0.55
```

B.1.11 socket_protocol.pure_json_protocol: No Callback at response instance for the request.

Testresult

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

Info	Send data, but no callback registered (pure_json_protocol).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
Send data: (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d 82 1c 8b 89
Receive data (79): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 7b 22 74 65 73 74 22 3a 20 22 74 65 73 74 22 7d 7d 82 1c 8b 89
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "{'test': 'test'}"
SJP: Received message with no registered callback. Sending negative response.
SJP: TX -> status: 1, service_id: 11, data_id: 45054, data: "None"
Send data: (67): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 64 61 74 61
↪  22 3a 20 6e 75 6c 6c 7d 24 86 3f 75
Receive data (67): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 64 61 74 61
↪  22 3a 20 6e 75 6c 6c 7d 24 86 3f 75
SJP: RX <- status: 1, service_id: 11, data_id: 45054, data: "None"
SJP: Received message has a peculiar status: Request has no callback. Data buffered.
SJP: Message data is stored in buffer and is now ready to be retrieved by receive method
```

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

```
Result (Return value of send method): True (<class 'bool'>)
Expectation (Return value of send method): result = True (<class 'bool'>)
```

Success	Response Status (Request has no callback. Data buffered.) transfered via pure_json_protocol is correct (Content 1 and Type is <class 'int'>).

```
Result (Response Status (Request has no callback. Data buffered.) transfered via
↪  pure_json_protocol): 1 (<class 'int'>)
Expectation (Response Status (Request has no callback. Data buffered.) transfered via
↪  pure_json_protocol): result = 1 (<class 'int'>)
```

Success	Response Data (no data) transfered via pure_json_protocol is correct (Content None and Type is <class 'NoneType'>).

```
Result (Response Data (no data) transfered via pure_json_protocol): None (<class 'NoneType'>)
Expectation (Response Data (no data) transfered via pure_json_protocol): result = None
↪  (<class 'NoneType'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<class 'NoneType'>)
```

B.1.12 socket_protocol.pure_json_protocol: Authentification required, but not processed/correctly processed.

Testresult

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

Info	Authentification with different secrets for request and response instance (pure_json_protocol).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Requesting seed for authentification
SJP: TX -> status: 0, service_id: 1, data_id: 0, data: "None"
Send data: (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75
↪  6c 6c 7d 9e 85 7b 8d
Receive data (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75
↪  6c 6c 7d 9e 85 7b 8d
SJP: RX <- status: 0, service_id: 1, data_id: 0, data: "None"
SJP: Executing callback __authentificate_create_seed__ to process received data
SJP: Got seed request, sending seed for authentification
SJP: TX -> status: 0, service_id: 2, data_id: 0, data:
↪  "'8492ca6edb9887708423c0282c2fb75d3b297a461f4fdee4dfec8a0f0028bc43'"
Send data: (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 32 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 38
↪  34 39 32 63 61 36 65 64 62 39 38 38 37 37 30 38 34 32 33 63 30 32 38 32 63 32 66 62 37 35
↪  64 33 62 32 39 37 61 34 36 31 66 34 66 64 65 65 34 64 66 65 63 38 61 30 66 30 30 32 38 62
↪  63 34 33 22 7d b3 21 58 a4
Receive data (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f
↪  69 64 22 3a 20 32 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22
↪  38 34 39 32 63 61 36 65 64 62 39 38 38 37 37 30 38 34 32 33 63 30 32 38 32 63 32 66 62 37
↪  35 64 33 62 32 39 37 61 34 36 31 66 34 66 64 65 65 34 64 66 65 63 38 61 30 66 30 30 32 38
↪  62 63 34 33 22 7d b3 21 58 a4
SJP: RX <- status: 0, service_id: 2, data_id: 0, data:
↪  "'8492ca6edb9887708423c0282c2fb75d3b297a461f4fdee4dfec8a0f0028bc43'"
SJP: Executing callback __authentificate_create_key__ to process received data
SJP: Got seed, sending key for authentification
SJP: TX -> status: 0, service_id: 3, data_id: 0, data:
↪  "'1b438e77e5310d5fdc5ad77d7bdef1879268dd4b4367f3243f73300f990ba7afcfc5ed5ad583261cee8dd68
↪  a46a591d8cabaa32bc182a0ad12470f7619fa4f15'"
Send data: (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f 69
↪  64 22 3a 20 33 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 31
↪  62 34 33 38 65 37 37 65 35 33 31 30 64 35 66 64 63 35 61 64 37 37 64 37 62 64 65 66 31 38
↪  37 39 32 36 38 64 64 34 62 34 33 36 37 66 33 32 34 33 66 37 33 33 30 30 66 39 39 30 62 61
↪  37 61 66 63 66 63 35 65 64 35 61 64 35 38 33 32 36 31 63 65 65 38 64 64 36 38 61 34 36 61
↪  35 39 31 64 38 63 61 62 61 61 33 32 62 63 31 38 32 61 30 61 64 31 32 34 37 30 66 37 36 31
↪  39 66 61 34 66 31 35 22 7d d4 d9 18 24
Receive data (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76 69 63 65 5f
↪  69 64 22 3a 20 33 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22
↪  31 62 34 33 38 65 37 37 65 35 33 31 30 64 35 66 64 63 35 61 64 37 37 64 37 62 64 65 66 31
↪  38 37 39 32 36 38 64 64 34 62 34 33 36 37 66 33 32 34 33 66 37 33 33 30 30 66 39 39 30 62
↪  61 37 61 66 63 66 63 35 65 64 35 61 64 35 38 33 32 36 31 63 65 65 38 64 64 36 38 61 34 36
↪  61 35 39 31 64 38 63 61 62 61 61 33 32 62 63 31 38 32 61 30 61 64 31 32 34 37 30 66 37 36
↪  31 39 66 61 34 66 31 35 22 7d d4 d9 18 24
SJP: RX <- status: 0, service_id: 3, data_id: 0, data:
↪  "'1b438e77e5310d5fdc5ad77d7bdef1879268dd4b4367f3243f73300f990ba7afcfc5ed5ad583261cee8dd68
↪  a46a591d8cabaa32bc182a0ad12470f7619fa4f15'"
SJP: Executing callback __authentificate_check_key__ to process received data
```

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

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

```
Result (Return value of send method): True (<class 'bool'>)
Expectation (Return value of send method): result = True (<class 'bool'>)
```

Success	Response Status (Authentification required) transfered via pure_json_protocol is correct (Content 2 and Type is <class 'int'>).

```
Result (Response Status (Authentification required) transfered via pure_json_protocol): 2
↪  (<class 'int'>)
Expectation (Response Status (Authentification required) transfered via pure_json_protocol):
↪  result = 2 (<class 'int'>)
```

Success	Response Data (no data) transfered via pure_json_protocol is correct (Content None and Type is <class 'NoneType'>).

```
Result (Response Data (no data) transfered via pure_json_protocol): None (<class 'NoneType'>)
Expectation (Response Data (no data) transfered via pure_json_protocol): result = None
↪  (<class 'NoneType'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<class 'NoneType'>)
```

B.1.13 socket_protocol.pure_json_protocol: Register a Callback which is already defined.

Testresult

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

Info	Registering two callbacks which overlap for at least one message (pure_json_protocol).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
```

Success	Exception (RegistrationError) detection variable is correct (Content True and Type is <class 'bool'>).

```
Result (Exception (RegistrationError) detection variable): True (<class 'bool'>)
Expectation (Exception (RegistrationError) detection variable): result = True (<class 'bool'>)
```

B.1.14 socket_protocol.pure_json_protocol: Authentification processed without secret.

Testresult

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

Info	Authentification with no secret definition (pure_json_protocol).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
```

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

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

B.1.15 socket_protocol.pure_json_protocol: Incompatible Callback return value(s).

Testresult

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

Info	Send and received data with incompatible callback (pure_json_protocol).

```
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: Cleaning up receive-buffer
SJP: Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT
SJP: TX -> status: 0, service_id: 10, data_id: 45054, data: "None"
Send data: (67): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 6e 75 6c 6c 7d e9 e9 77 c0
Receive data (67): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 35 30 35 34 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 6e 75 6c 6c 7d e9 e9 77 c0
SJP: RX <- status: 0, service_id: 10, data_id: 45054, data: "None"
SJP: Executing callback response_data_method_fail to process received data
SJP: TX -> status: 0, service_id: 10, data_id: 48879, data: "None"
Send data: (67): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 6e 75 6c 6c 7d da 63 1a 84
Receive data (67): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪  22 3a 20 6e 75 6c 6c 7d da 63 1a 84
SJP: RX <- status: 0, service_id: 10, data_id: 48879, data: "None"
SJP: Received message with no registered callback. Sending negative response.
SJP: TX -> status: 1, service_id: 11, data_id: 48879, data: "None"
Send data: (67): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 64 61 74 61
↪  22 3a 20 6e 75 6c 6c 7d 17 0c 52 31
Receive data (67): 7b 22 64 61 74 61 5f 69 64 22 3a 20 34 38 38 37 39 2c 20 22 73 65 72 76 69
↪  63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 64 61 74 61
↪  22 3a 20 6e 75 6c 6c 7d 17 0c 52 31
SJP: RX <- status: 1, service_id: 11, data_id: 48879, data: "None"
SJP: Received message has a peculiar status: Request has no callback. Data buffered.
SJP: Executing callback response_data_method_fail to process received data
```

Success	Exception (TypeError) detection variable is correct (Content True and Type is <class 'bool'>).

```
Result (Exception (TypeError) detection variable): True (<class 'bool'>)
Expectation (Exception (TypeError) detection variable): result = True (<class 'bool'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 45054) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xaffe is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 45054) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xaffe): None (<class 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xaffe): result = None (<class 'NoneType'>)
```

Success	Exception (TypeError) detection variable is correct (Content True and Type is <class 'bool'>).

```
Result (Exception (TypeError) detection variable): True (<class 'bool'>)
Expectation (Exception (TypeError) detection variable): result = True (<class 'bool'>)
```

Success	Return Value (request instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 11; data_id: 48879) not in buffer.
Result (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<class 'NoneType'>)
Expectation (Return Value (request instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): result = None (<class 'NoneType'>)
```

Success	Return Value (response instance) for pure_json_protocol.receive with data data_id 0xbeef is correct (Content None and Type is <class 'NoneType'>).

```
SJP: TIMEOUT (0.1s): Requested data (service_id: 10; data_id: 48879) not in buffer.
Result (Return Value (response instance) for pure_json_protocol.receive with data data_id
↪  0xbeef): None (<class 'NoneType'>)
Expectation (Return Value (response instance) for pure_json_protocol.receive with data
↪  data_id 0xbeef): result = None (<class 'NoneType'>)
```

C Test-Coverage

C.1 socket_protocol

The line coverage for socket_protocol was 97.8%
The branch coverage for socket_protocol was 96.8%

C.1.1 socket_protocol.__init__.py

The line coverage for socket_protocol.__init__.py was 97.8%
The branch coverage for socket_protocol.__init__.py was 96.8%

```
#!/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.struct_json_protocol`
* :class:`socket_protocol.pure_json_protocol`

**Unittest:**

        See also the :download:`unittest <../../socket_protocol/_testresults_/unittest.pdf>`
    documentation.
"""
__DEPENDENCIES__ = ['stringtools']

import stringtools

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


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


__DESCRIPTION__ = """The Module {\\tt %s} is designed to pack and unpack data for serial
    transportation.
For more Information read the sphinx documentation.""" % __name__.replace('_', '\_')
"""The Module Description"""
__INTERPRETER__ = (2, 3)
"""The Tested Interpreter-Versions"""


class RegistrationError(BaseException):
    pass
```

```
55
56
57  class callback_storage(dict):
58      def __init__(self):
59          dict.__init__(self)
60
61      def get(self, service_id, data_id):
62          if service_id is not None and data_id is not None:
63              try:
64                  return self[service_id][data_id]
65              except KeyError:
66                  pass  # nothing to append
67          if data_id is not None:
68              try:
69                  return self[None][data_id]
70              except KeyError:
71                  pass  # nothing to append
72          if service_id is not None:
73              try:
74                  return self[service_id][None]
75              except KeyError:
76                  pass  # nothing to append
77          try:
78              return self[None][None]
79          except KeyError:
80              pass  # nothing to append
81          return (None, None, None)
82
83      def add(self, service_id, data_id, callback, *args, **kwargs):
84          if self.get(service_id, data_id) != (None, None, None):
85              raise RegistrationError("Callback for service_id (%s) and data_id (%s) already exists
    " % (repr(service_id), repr(data_id)))
86          if service_id not in self:
87              self[service_id] = {}
88          self[service_id][data_id] = (callback, args, kwargs)
89
90
91  class data_storage(dict):
92      KEY_STATUS = 'status'
93      KEY_SERVICE_ID = 'service_id'
94      KEY_DATA_ID = 'data_id'
95      KEY_DATA = 'data'
96
97      def __init__(self, *args, **kwargs):
98          dict.__init__(self, *args, **kwargs)
99
100     def get_status(self, default=None):
101         return self.get(self.KEY_STATUS, default)
102
103     def get_service_id(self, default=None):
104         return self.get(self.KEY_SERVICE_ID, default)
105
106     def get_data_id(self, default=None):
107         return self.get(self.KEY_DATA_ID, default)
108
109     def get_data(self, default=None):
110         return self.get(self.KEY_DATA, default)
111
112
113 class struct_json_protocol(object):
```

```
114     """
115     :param comm_instance: a communication instance supportin at least these functions: :func:`
        register_callback`, :func:`register_disconnect_callback`, :func:`send`.
116     :type comm_instance: instance
117     :param secret: A secret (e.g. created by ``binascii.hexlify(os.urandom(24))``).
118     :type secret: str
119
120     This communication protocol supports to transfer a Service-ID, Data-ID and Data. The
        transmitted data is shorter than :class:`pure_json_protocol`.
121
122     .. note::
123         This class is here for compatibility reasons. Usage of :class:`pure_json_protocol` is
        recommended.
124
125     **Example:**
126
127     Server:
128
129     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__struct_json_protocol_server.py
130
131     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__struct_json_protocol_server.log
132
133
134     Client:
135
136     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__struct_json_protocol_client.py
137
138     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__struct_json_protocol_client.log
139     """
140     LOG_PREFIX = 'SJP:'
141
142     SID_AUTH_SEED_REQUEST = 1
143     SID_AUTH_KEY_REQUEST = 2
144     SID_AUTH_KEY_CHECK_REQUEST = 3
145     SID_AUTH_KEY_CHECK_RESPONSE = 4
146     SID_READ_REQUEST = 10
147     SID_READ_RESPONSE = 11
148     SID_WRITE_REQUEST = 20
149     SID_WRITE_RESPONSE = 21
150     SID_EXECUTE_REQUEST = 30
151     SID_EXECUTE_RESPONSE = 31
152
153     SID_RESPONSE_DICT = {SID_AUTH_SEED_REQUEST: SID_AUTH_KEY_REQUEST,
154                          SID_AUTH_KEY_REQUEST: SID_AUTH_KEY_CHECK_REQUEST,
155                          SID_AUTH_KEY_CHECK_REQUEST: SID_AUTH_KEY_CHECK_RESPONSE,
156                          SID_READ_REQUEST: SID_READ_RESPONSE,
157                          SID_WRITE_REQUEST: SID_WRITE_RESPONSE,
158                          SID_EXECUTE_REQUEST: SID_EXECUTE_RESPONSE}
159
160     SID_AUTH_LIST = [SID_AUTH_SEED_REQUEST, SID_AUTH_KEY_REQUEST, SID_AUTH_KEY_CHECK_REQUEST,
        SID_AUTH_KEY_CHECK_RESPONSE]
161
162     STATUS_OKAY = 0
163     STATUS_BUFFERING_UNHANDLED_REQUEST = 1
164     STATUS_AUTH_REQUIRED = 2
165     STATUS_SERVICE_OR_DATA_UNKNOWN = 3
166     STATUS_CHECKSUM_ERROR = 4
167     STATUS_OPERATION_NOT_PERMITTED = 5
168     STATUS_NAMES = {STATUS_OKAY: 'Okay',
```

```
169                     STATUS_BUFFERING_UNHANDLED_REQUEST: 'Request has no callback. Data buffered.'
    ,
170                     STATUS_AUTH_REQUIRED: 'Authentification required',
171                     STATUS_SERVICE_OR_DATA_UNKNOWN: 'Service or Data unknown',
172                     STATUS_CHECKSUM_ERROR: 'Checksum Error',
173                     STATUS_OPERATION_NOT_PERMITTED: 'Operation not permitted'}
174
175     AUTH_STATE_UNKNOWN_CLIENT = 0
176     AUTH_STATE_SEED_REQUESTED = 1
177     AUTH_STATE_SEED_TRANSFERRED = 2
178     AUTH_STATE_KEY_TRANSFERRED = 3
179     AUTH_STATE_TRUSTED_CLIENT = 4
180     AUTH_STATUS_NAMES = {AUTH_STATE_UNKNOWN_CLIENT: 'Unknown Client',
181                          AUTH_STATE_SEED_REQUESTED: 'Seed was requested',
182                          AUTH_STATE_SEED_TRANSFERRED: 'Seed has been sent',
183                          AUTH_STATE_KEY_TRANSFERRED: 'Key has been sent',
184                          AUTH_STATE_TRUSTED_CLIENT: 'Trusted Client'}
185
186     def __init__(self, comm_instance, secret=None, auto_auth=False):
187         self.__secret__ = secret
188         self.__auto_auth__ = auto_auth
189         self.__clean_receive_buffer__()
190         self.__callbacks__ = callback_storage()
191         self.__callbacks__.add(self.SID_AUTH_SEED_REQUEST, 0, self.__authentificate_create_seed__
    )
192         self.__callbacks__.add(self.SID_AUTH_KEY_REQUEST, 0, self.__authentificate_create_key__)
193         self.__callbacks__.add(self.SID_AUTH_KEY_CHECK_REQUEST, 0, self.
    __authentificate_check_key__)
194         self.__callbacks__.add(self.SID_AUTH_KEY_CHECK_RESPONSE, 0, self.
    __authentificate_process_feedback__)
195         self.__authentification_state_reset__()
196         self.__seed__ = None
197         self.__comm_inst__ = comm_instance
198         self.__comm_inst__.register_callback(self.__data_available_callback__)
199         self.__comm_inst__.register_connect_callback(self.__connection_established__)
200         self.__comm_inst__.register_disconnect_callback(self.__authentification_state_reset__)
201
202     def connected(self):
203         return self.__comm_inst__.is_connected()
204
205     def connection_established(self):
206         return self.connected() and (self.__secret__ is None or self.check_authentification_state
    ())
207
208     def reconnect(self):
209         return self.__comm_inst__.reconnect()
210
211     def __connection_established__(self):
212         self.__clean_receive_buffer__()
213         if self.__auto_auth__ and self.__comm_inst__.IS_CLIENT and self.__secret__ is not None:
214             self.authentificate()
215
216     def __authentification_state_reset__(self):
217         logger.info("%s Resetting authentification state to AUTH_STATE_UNKNOWN_CLIENT", self.
    LOG_PREFIX)
218         self.__authentification_state__ = self.AUTH_STATE_UNKNOWN_CLIENT
219
220     def __analyse_frame__(self, frame):
221         status, service_id, data_id = struct.unpack('>III', frame[0:12])
222         if sys.version_info >= (3, 0):
223             data = json.loads(frame[12:-1].decode('utf-8'))
224         else:
```

```
225             data = json.loads(frame[12:-1])
226         return self.__mk_msg__(status, service_id, data_id, data)
227 
228     def __build_frame__(self, service_id, data_id, data, status=STATUS_OKAY):
229         frame = struct.pack('>III', status, service_id, data_id)
230         if sys.version_info >= (3, 0):
231             frame += bytes(json.dumps(data), 'utf-8')
232             frame += self.__calc_chksum__(frame)
233         else:
234             frame += json.dumps(data)
235             frame += self.__calc_chksum__(frame)
236         return frame
237 
238     def __calc_chksum__(self, raw_data):
239         chksum = 0
240         for b in raw_data:
241             if sys.version_info >= (3, 0):
242                 chksum ^= b
243             else:
244                 chksum ^= ord(b)
245         if sys.version_info >= (3, 0):
246             return bytes([chksum])
247         else:
248             return chr(chksum)
249 
250     def __check_frame_checksum__(self, frame):
251         return self.__calc_chksum__(frame[:-1]) == frame[-1:]
252 
253     def __data_available_callback__(self, comm_inst):
254         frame = comm_inst.receive()
255         if not self.__check_frame_checksum__(frame):
256             logger.warning("%s Received message has a wrong checksum and will be ignored: %s.",
    self.LOG_PREFIX, stringtools.hexlify(frame))
257         else:
258             msg = self.__analyse_frame__(frame)
259             logger.info(
260                 '%s RX <- status: %s, service_id: %s, data_id: %s, data: "%s"',
261                 self.LOG_PREFIX,
262                 repr(msg.get_status()),
263                 repr(msg.get_service_id()),
264                 repr(msg.get_data_id()),
265                 repr(msg.get_data())
266             )
267             callback, args, kwargs = self.__callbacks__.get(msg.get_service_id(), msg.get_data_id
    ())
268             if msg.get_service_id() in self.SID_RESPONSE_DICT.keys():
269                 #
270                 # REQUEST RECEIVED
271                 #
272                 if self.__secret__ is not None and not self.check_authentification_state() and
    msg.get_service_id() not in self.SID_AUTH_LIST:
273                     status = self.STATUS_AUTH_REQUIRED
274                     data = None
275                     logger.warning("%s Received message needs authentification: %s. Sending
    negative response.", self.LOG_PREFIX, self.AUTH_STATUS_NAMES.get(self.
    __authentification_state__, 'Unknown authentification status!'))
276                 elif callback is None:
277                     logger.warning("%s Received message with no registered callback. Sending
    negative response.", self.LOG_PREFIX)
278                     status = self.STATUS_BUFFERING_UNHANDLED_REQUEST
279                     data = None
280                 else:
```

```
281                     try:
282                         logger.debug("%s Executing callback %s to process received data", self.
    LOG_PREFIX, callback.__name__)
283                         status, data = callback(msg, *args, **kwargs)
284                     except TypeError:
285                         raise TypeError('Check return value of callback function {callback_name}
    for service_id  {service_id} and data_id {data_id}'.format(callback_name=callback.__name__,
    service_id=repr(msg.get_service_id()), data_id=repr(msg.get_data_id())))
286                 self.send(self.SID_RESPONSE_DICT[msg.get_service_id()], msg.get_data_id(), data,
    status=status)
287             else:
288                 #
289                 # RESPONSE RECEIVED
290                 #
291                 if msg.get_status() not in [self.STATUS_OKAY]:
292                     logger.warning("%s Received message has a peculiar status: %s", self.
    LOG_PREFIX, self.STATUS_NAMES.get(msg.get_status(), 'Unknown status response!'))
293                 if callback is None:
294                     status = self.STATUS_OKAY
295                     data = None
296                     self.__buffer_received_data__(msg)
297                 else:
298                     try:
299                         logger.debug("%s Executing callback %s to process received data", self.
    LOG_PREFIX, callback.__name__)
300                         status, data = callback(msg, *args, **kwargs)
301                     except TypeError:
302                         raise TypeError('Check return value of callback function {callback_name}
    for service_id  {service_id} and data_id {data_id}'.format(callback_name=callback.__name__,
    service_id=repr(msg.get_service_id()), data_id=repr(msg.get_data_id())))
303
304     def __buffer_received_data__(self, msg):
305         if not msg.get_service_id() in self.__msg_buffer__:
306             self.__msg_buffer__[msg.get_service_id()] = {}
307         if not msg.get_data_id() in self.__msg_buffer__[msg.get_service_id()]:
308             self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()] = []
309         self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()].append(msg)
310         logger.debug("%s Message data is stored in buffer and is now ready to be retrieved by
    receive method", self.LOG_PREFIX)
311
312     def __clean_receive_buffer__(self):
313         logger.debug("%s Cleaning up receive-buffer", self.LOG_PREFIX)
314         self.__msg_buffer__ = {}
315
316     def receive(self, service_id, data_id, timeout=1):
317         data = None
318         cnt = 0
319         while data is None and cnt < timeout * 10:
320             try:
321                 data = self.__msg_buffer__.get(service_id, {}).get(data_id, []).pop(0)
322             except IndexError:
323                 data = None
324             cnt += 1
325             time.sleep(0.1)
326         if data is None and cnt >= timeout * 10:
327             logger.warning('%s TIMEOUT (%ss): Requested data (service_id: %s; data_id: %s) not in
     buffer.', self.LOG_PREFIX, repr(timeout), repr(service_id), repr(data_id))
328         return data
329
330     def __mk_msg__(self, status, service_id, data_id, data):
331         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})
```

```
332
333     def send(self, service_id, data_id, data, status=STATUS_OKAY, timeout=2, log_lvl=logging.INFO
        ):
334         """
335         :param service_id: The Service-ID for the message. See class definitions starting with ``
        SERVICE_``.
336         :type service_id: int
337         :param data_id: The Data-ID for the message.
338         :type data_id: int
339         :param data: The data to be transfered. The data needs to be json compatible.
340         :type data: str
341         :param status: The Status for the message. All requests should have ``STATUS_OKAY``.
342         :type status: int
343         :param timeout: The timeout for sending data (e.g. time to establish new connection).
344         :type timeout: float
345         :param rx_log_lvl: The log level to log outgoing TX-data
346         :type rx_log_lvl: int
347         :return: True if data had been sent, otherwise False.
348         :rtype: bool
349
350         This methods sends out a message with the given content.
351         """
352         logger.log(log_lvl, '%s TX -> status: %d, service_id: %d, data_id: %d, data: "%s"', self.
        LOG_PREFIX, status, service_id, data_id, repr(data))
353         return self.__comm_inst__.send(self.__build_frame__(service_id, data_id, data, status),
        timeout=timeout, log_lvl=logging.DEBUG)
354
355     def register_callback(self, service_id, data_id, callback, *args, **kwargs):
356         """
357         :param service_id: The Service-ID for the message. See class definitions starting with ``
        SID_``.
358         :type service_id: int
359         :param data_id: The Data-ID for the message.
360         :type data_id: int
361         :returns: True, if registration was successfull; False, if registration failed (e.g.
        existance of a callback for this configuration)
362         :rtype: bool
363
364         This method registers a callback for the given parameters. Givin ``None`` means, that all
         Service-IDs or all Data-IDs are used.
365         If a message hitting these parameters has been received, the callback will be executed.
366
367         .. note:: The :func:`callback` is priorised in the following order:
368
369             * Callbacks with defined Service-ID and Data-ID.
370             * Callbacks with a defined Data-ID.
371             * Callbacks with a defined Service-ID.
372             * Unspecific Callbacks
373
374         .. note:: The :func:`callback` is executed with these arguments:
375
376             :param msg: A :class:`dict` containing all message information.
377             :returns: status (see class definition starting with ``STATUS_``), response_data (
        JSON compatible object)
378         """
379         self.__callbacks__.add(service_id, data_id, callback, *args, **kwargs)
380
381     def authentificate(self, timeout=2):
382         """
383         :param timeout: The timeout for the authentification (requesting seed, sending key and
        getting authentification_feedback).
```

```
384         :type timeout: float
385         :returns: True, if authentification was successfull; False, if not.
386         :rtype: bool
387 
388         This method authetificates the client at the server.
389 
390         .. note:: An authentification will only processed, if a secret had been given on
    initialisation.
391 
392         .. note:: Client and Server needs to use the same secret.
393         """
394         if self.__secret__ is not None:
395             self.__authentification_state__ = self.AUTH_STATE_SEED_REQUESTED
396             logger.info("%s Requesting seed for authentification", self.LOG_PREFIX)
397             self.send(self.SID_AUTH_SEED_REQUEST, 0, None)
398             cnt = 0
399             while cnt < timeout * 10:
400                 time.sleep(0.1)
401                 if self.__authentification_state__ == self.AUTH_STATE_TRUSTED_CLIENT:
402                     return True
403                 elif self.__authentification_state__ == self.AUTH_STATE_UNKNOWN_CLIENT:
404                     break
405                 cnt += 1
406         return False
407 
408     def check_authentification_state(self):
409         """
410         :return: True, if authentification state is okay, otherwise False
411         :rtype: bool
412         """
413         return self.__authentification_state__ == self.AUTH_STATE_TRUSTED_CLIENT
414 
415     def __authentificate_salt_and_hash__(self, seed):
416         if sys.version_info >= (3, 0):
417             return hashlib.sha512(bytes(seed, 'utf-8') + self.__secret__).hexdigest()
418         else:
419             return hashlib.sha512(seed.encode('utf-8') + self.__secret__.encode('utf-8')).
    hexdigest()
420 
421     def __authentificate_create_seed__(self, msg):
422         logger.info("%s Got seed request, sending seed for authentification", self.LOG_PREFIX)
423         self.__authentification_state__ = self.AUTH_STATE_SEED_TRANSFERRED
424         if sys.version_info >= (3, 0):
425             self.__seed__ = binascii.hexlify(os.urandom(32)).decode('utf-8')
426         else:
427             self.__seed__ = binascii.hexlify(os.urandom(32))
428         return self.STATUS_OKAY, self.__seed__
429 
430     def __authentificate_create_key__(self, msg):
431         logger.info("%s Got seed, sending key for authentification", self.LOG_PREFIX)
432         self.__authentification_state__ = self.AUTH_STATE_KEY_TRANSFERRED
433         seed = msg.get_data()
434         key = self.__authentificate_salt_and_hash__(seed)
435         return self.STATUS_OKAY, key
436 
437     def __authentificate_check_key__(self, msg):
438         key = msg.get_data()
439         if key == self.__authentificate_salt_and_hash__(self.__seed__):
440             self.__authentification_state__ = self.AUTH_STATE_TRUSTED_CLIENT
441             logger.info("%s Got correct key, sending positive authentification feedback", self.
    LOG_PREFIX)
442             return self.STATUS_OKAY, True
```

```
443         else:
444             self.__authentification_state__ = self.AUTH_STATE_UNKNOWN_CLIENT
445             logger.info("%s Got incorrect key, sending negative authentification feedback", self.
    LOG_PREFIX)
446             return self.STATUS_OKAY, False
447 
448     def __authentificate_process_feedback__(self, msg):
449         feedback = msg.get_data()
450         if feedback:
451             self.__authentification_state__ = self.AUTH_STATE_TRUSTED_CLIENT
452             logger.info("%s Got positive authentification feedback", self.LOG_PREFIX)
453         else:
454             self.__authentification_state__ = self.AUTH_STATE_UNKNOWN_CLIENT
455             logger.warning("%s Got negative authentification feedback", self.LOG_PREFIX)
456         return self.STATUS_OKAY, None
457 
458 
459 class pure_json_protocol(struct_json_protocol):
460     """
461     :param comm_instance: a communication instance supportin at least these functions: :func:`
    register_callback`, :func:`register_disconnect_callback`, :func:`send`.
462     :type comm_instance: instance
463     :param secret: A secret (e.g. created by ``binascii.hexlify(os.urandom(24))``).
464     :type secret: str
465 
466     This communication protocol supports to transfer a Service-ID, Data-ID and Data.
467 
468     **Example:**
469 
470     Server:
471 
472     .. literalinclude:: ../../socket_protocol/_examples_/
    socket_protocol__pure_json_protocol_server.py
473 
474     .. literalinclude:: ../../socket_protocol/_examples_/
    socket_protocol__pure_json_protocol_server.log
475 
476 
477     Client:
478 
479     .. literalinclude:: ../../socket_protocol/_examples_/
    socket_protocol__pure_json_protocol_client.py
480 
481     .. literalinclude:: ../../socket_protocol/_examples_/
    socket_protocol__pure_json_protocol_client.log
482     """
483     def __init__(self, *args, **kwargs):
484         struct_json_protocol.__init__(self, *args, **kwargs)
485 
486     def __build_frame__(self, service_id, data_id, data, status=struct_json_protocol.STATUS_OKAY)
    :
487         data_frame = json.dumps(self.__mk_msg__(status, service_id, data_id, data))
488         if sys.version_info >= (3, 0):
489             data_frame = bytes(data_frame, 'utf-8')
490         checksum = self.__calc_chksum__(data_frame)
491         return data_frame + checksum
492 
493     def __analyse_frame__(self, frame):
494         if sys.version_info >= (3, 0):
495             return data_storage(json.loads(frame[:-4].decode('utf-8')))
496         else:
497             return data_storage(json.loads(frame[:-4]))
```

```
498
499     def __calc_chksum__(self, raw_data):
500         return struct.pack('>I', binascii.crc32(raw_data) & 0xffffffff)
501
502     def __check_frame_checksum__(self, frame):
503         return self.__calc_chksum__(frame[:-4]) == frame[-4:]
```