Unittest for socket_protocol

January 29, 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	44bfc23658f5a000bcabcf2a34875620
Dependencies	
stringtools	88a3eed174bd2239a6c1d928081e5b6d

1.2 Unittest Information

Unittest Information	
Version	cd82b7d4eb571a53181f2cb9c7b37417
Testruns with	python 2.7.17 (final), python 3.6.9 (final)

1.3 Test System Information

System Information	
Architecture	64bit
Distribution	LinuxMint 19.3 tricia
Hostname	ahorn
Kernel	5.3.0-28-generic (#30 18.04.1-Ubuntu SMP Fri Jan 17 06:14:09 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 2.7.17 (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	11.291s

2.2 Test-Statistic for testrun with python 3.6.9 (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	11.335s

2.3 Coverage Statistic

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

3 Testcases with no corresponding Requirement

3.1 Summary for testrun with python 2.7.17 (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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (44)
Start-Time:	2020-01-29 07:23:18,047
Finished-Time:	2020-01-29 07:23:18,049
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 <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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (42)
Start-Time:	2020-01-29 07:23:16,632
Finished-Time:	2020-01-29 07:23:18,044
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 <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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (36)
Start-Time:	2020-01-29 07:23:14,012
Finished-Time:	2020-01-29 07:23:14,516
Time-Consumption	0.504s

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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (45)
Start-Time:	2020-01-29 07:23:18,049
Finished-Time:	2020-01-29 07:23:18,461
Time-Consumption	0.412s

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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (41)
Start-Time:	2020-01-29 07:23:15,924

Finished-Time:	2020-01-29 07:23:16,632
Time-Consumption	0.708s

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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (43)
Start-Time:	2020-01-29 07:23:18,045
Finished-Time:	2020-01-29 07:23:18,046
Time-Consumption	0.002s

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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (31)
Start-Time:	2020-01-29 07:23:12,096
Finished-Time:	2020-01-29 07:23:12,802
Time-Consumption	0.706s

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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (27)
Start-Time:	2020-01-29 07:23:08,575
Finished-Time:	2020-01-29 07:23:09,981
Time-Consumption	1.406s

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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (26)
Start-Time:	2020-01-29 07:23:07,870
Finished-Time:	2020-01-29 07:23:08,574
Time-Consumption	0.704s

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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (37)
Start-Time:	2020-01-29 07:23:14,516
Finished-Time:	2020-01-29 07:23:15,924
Time-Consumption	1.408s

Testsummary:

Success	Timeout for authentification is correct (Content 0.20073795318603516 in [0.2 ... 0.22000000000000003] and Type is <type 'float'>).
Success	Timeout for authentification is correct (Content 0.5015730857849121 in [0.5 ... 0.55] and Type is <type 'float'>).
Success	Timeout for send method is correct (Content 0.20078516006469727 in [0.2 ... 0.22000000000000003] and Type is <type 'float'>).
Success	Timeout for send method is correct (Content 0.501331090927124 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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (30)
Start-Time:	2020-01-29 07:23:11,392
Finished-Time:	2020-01-29 07:23:12,096
Time-Consumption	0.704s

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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (28)
Start-Time:	2020-01-29 07:23:09,981
Finished-Time:	2020-01-29 07:23:10,687
Time-Consumption	0.706s

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

Start-Time:	2020-01-29 07:23:10,687
Finished-Time:	2020-01-29 07:23:11,391
Time-Consumption	0.704s

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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (32)
Start-Time:	2020-01-29 07:23:12,802
Finished-Time:	2020-01-29 07:23:14,012
Time-Consumption	1.209s

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 2.7.17 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (25)
Start-Time:	2020-01-29 07:23:07,166
Finished-Time:	2020-01-29 07:23:07,870
Time-Consumption	0.704s

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 3.6.9 (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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (44)
Start-Time:	2020-01-29 07:23:29,909
Finished-Time:	2020-01-29 07:23:29,910
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 <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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (42)
Start-Time:	2020-01-29 07:23:28,483
Finished-Time:	2020-01-29 07:23:29,907
Time-Consumption	1.424s

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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (36)
Start-Time:	2020-01-29 07:23:25,848
Finished-Time:	2020-01-29 07:23:26,359
Time-Consumption	0.512s

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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (45)
Start-Time:	2020-01-29 07:23:29,911

Finished-Time:	2020-01-29 07:23:30,325
Time-Consumption	0.414s

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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (41)
Start-Time:	2020-01-29 07:23:27,766
Finished-Time:	2020-01-29 07:23:28,483
Time-Consumption	0.717s

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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (43)
Start-Time:	2020-01-29 07:23:29,908
Finished-Time:	2020-01-29 07:23:29,909

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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (31)
Start-Time:	2020-01-29 07:23:23,934
Finished-Time:	2020-01-29 07:23:24,639
Time-Consumption	0.705s

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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (27)
Start-Time:	2020-01-29 07:23:20,400
Finished-Time:	2020-01-29 07:23:21,810
Time-Consumption	1.409s

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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (26)
Start-Time:	2020-01-29 07:23:19,692
Finished-Time:	2020-01-29 07:23:20,400
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 <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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (37)

Start-Time:	2020-01-29 07:23:26,359
Finished-Time:	2020-01-29 07:23:27,766
Time-Consumption	1.406s

Testsummary:

Success	Timeout for authentification is correct (Content 0.2016153335571289 in [0.2 ... 0.22000000000000003] and Type is <class 'float'>).
Success	Timeout for authentification is correct (Content 0.5011227130889893 in [0.5 ... 0.55] and Type is <class 'float'>).
Success	Timeout for send method is correct (Content 0.20057988166809082 in [0.2 ... 0.22000000000000003] and Type is <class 'float'>).
Success	Timeout for send method is correct (Content 0.5010545253753662 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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (30)
Start-Time:	2020-01-29 07:23:23,228
Finished-Time:	2020-01-29 07:23:23,934
Time-Consumption	0.706s

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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (28)
Start-Time:	2020-01-29 07:23:21,810

Finished-Time:	2020-01-29 07:23:22,518
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 <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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (29)
Start-Time:	2020-01-29 07:23:22,518
Finished-Time:	2020-01-29 07:23:23,227
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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (32)
Start-Time:	2020-01-29 07:23:24,640
Finished-Time:	2020-01-29 07:23:25,847
Time-Consumption	1.208s

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 3.6.9 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (25)
Start-Time:	2020-01-29 07:23:18,985
Finished-Time:	2020-01-29 07:23:19,692
Time-Consumption	0.707s

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 2.7.17 (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:
↪  "'0cdb40d33d9d2821ef18e2336f30f235cd5b936f666c1470c2b24278ec4fae97'"
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 63 64 62 34 30 64 33 33 64 39 64 32 38 32
↪  31 65 66 31 38 65 32 33 33 36 66 33 30 66 32 33 35 63 64 35 62 39 33 36 66 36 36 36 63 31
↪  34 37 30 63 32 62 32 34 32 37 38 65 63 34 66 61 65 39 37 22 2c 20 22 64 61 74 61 5f 69 64
↪  22 3a 20 30 7d 47 0c 2d 0c
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 63 64 62 34 30 64 33 33 64 39 64 32 38
↪  32 31 65 66 31 38 65 32 33 33 36 66 33 30 66 32 33 35 63 64 35 62 39 33 36 66 36 36 36 63
↪  31 34 37 30 63 32 62 32 34 32 37 38 65 63 34 66 61 65 39 37 22 2c 20 22 64 61 74 61 5f 69
↪  64 22 3a 20 30 7d 47 0c 2d 0c
SJP: RX <- status: 0, service_id: 2, data_id: 0, data:
↪  "u'0cdb40d33d9d2821ef18e2336f30f235cd5b936f666c1470c2b24278ec4fae97'"
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:
↪  "'a3847d3f2d7f0791ce459dcf0b5ae84ba744080e51bd84d485c9732df2052b7d673eb45768ba85abbee42f3
↪  7bb98b9e8c8f81560eb0d9d785ab5bf190b19c314'"
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 61 33 38 34 37 64 33 66 32 64 37 66 30 37 39
↪  31 63 65 34 35 39 64 63 66 30 62 35 61 65 38 34 62 61 37 34 34 30 38 30 65 35 31 62 64 38
↪  34 64 34 38 35 63 39 37 33 32 64 66 32 30 35 32 62 37 64 36 37 33 65 62 34 35 37 36 38 62
↪  61 38 35 61 62 62 65 65 34 32 66 33 37 62 62 39 38 62 39 65 38 63 38 66 38 31 35 36 30 65
↪  62 30 64 39 64 37 38 35 61 62 35 62 66 31 39 30 62 31 39 63 33 31 34 22 2c 20 22 64 61 74
↪  61 5f 69 64 22 3a 20 30 7d 83 91 71 cc
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 61 33 38 34 37 64 33 66 32 64 37 66 30 37
↪  39 31 63 65 34 35 39 64 63 66 30 62 35 61 65 38 34 62 61 37 34 34 30 38 30 65 35 31 62 64
↪  38 34 64 34 38 35 63 39 37 33 32 64 66 32 30 35 32 62 37 64 36 37 33 65 62 34 35 37 36 38
↪  62 61 38 35 61 62 62 65 65 34 32 66 33 37 62 62 39 38 62 39 65 38 63 38 66 38 31 35 36 30
↪  65 62 30 64 39 64 37 38 35 61 62 35 62 66 31 39 30 62 31 39 63 33 31 34 22 2c 20 22 64 61
↪  74 61 5f 69 64 22 3a 20 30 7d 83 91 71 cc
SJP: RX <- status: 0, service_id: 3, data_id: 0, data:
↪  "u'a3847d3f2d7f0791ce459dcf0b5ae84ba744080e51bd84d485c9732df2052b7d673eb45768ba85abbee42f
↪  37bb98b9e8c8f81560eb0d9d785ab5bf190b19c314'"
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 0.20073795318603516 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): 0.20073795318603516 (<type 'float'>)
Expectation (Timeout for authentification): 0.2 <= result <= 0.22000000000000003
```

Success	Timeout for authentification is correct (Content 0.5015730857849121 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): 0.5015730857849121 (<type 'float'>)
Expectation (Timeout for authentification): 0.5 <= result <= 0.55
```

Success	Timeout for send method is correct (Content 0.20078516006469727 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): 0.20078516006469727 (<type 'float'>)
Expectation (Timeout for send method): 0.2 <= result <= 0.22000000000000003
```

Success	Timeout for send method is correct (Content 0.501331090927124 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): 0.501331090927124 (<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:
↪  "'da7700cc13defc6cb1edbf070d870e33e6c7de100ed3b618ea764b9f8f1b7ef2'"
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 64 61 37 37 30 30 63 63 31 33 64 65 66 63 36
↪  63 62 31 65 64 62 66 30 37 30 64 38 37 30 65 33 33 65 36 63 37 64 65 31 30 30 65 64 33 62
↪  36 31 38 65 61 37 36 34 62 39 66 38 66 31 62 37 65 66 32 22 2c 20 22 64 61 74 61 5f 69 64
↪  22 3a 20 30 7d 16 95 fe 2e
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 64 61 37 37 30 30 63 63 31 33 64 65 66 63
↪  36 63 62 31 65 64 62 66 30 37 30 64 38 37 30 65 33 33 65 36 63 37 64 65 31 30 30 65 64 33
↪  62 36 31 38 65 61 37 36 34 62 39 66 38 66 31 62 37 65 66 32 22 2c 20 22 64 61 74 61 5f 69
↪  64 22 3a 20 30 7d 16 95 fe 2e
SJP: RX <- status: 0, service_id: 2, data_id: 0, data:
↪  "u'da7700cc13defc6cb1edbf070d870e33e6c7de100ed3b618ea764b9f8f1b7ef2'"
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:
↪  "'d5d1672dedcccda1026a7f34ab5ae8bec00e68f8583e8dc6bc77a60363fff0c37589e1b6127c7a52e47fa8c
↪  5c0e34041848ce4950a744ce91f6b3a508ffd1c8e'"
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 64 35 64 31 36 37 32 64 65 64 63 63 63 64 61
↪  31 30 32 36 61 37 66 33 34 61 62 35 61 65 38 62 65 63 30 30 65 36 38 66 38 35 38 33 65 38
↪  64 63 36 62 63 37 37 61 36 30 33 36 33 66 66 66 30 63 33 37 35 38 39 65 31 62 36 31 32 37
↪  63 37 61 35 32 65 34 37 66 61 38 63 35 63 30 65 33 34 30 34 31 38 34 38 63 65 34 39 35 30
↪  61 37 34 34 63 65 39 31 66 36 62 33 61 35 30 38 66 66 64 31 63 38 65 22 2c 20 22 64 61 74
↪  61 5f 69 64 22 3a 20 30 7d 39 b3 59 ab
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 64 35 64 31 36 37 32 64 65 64 63 63 63 64
↪  61 31 30 32 36 61 37 66 33 34 61 62 35 61 65 38 62 65 63 30 30 65 36 38 66 38 35 38 33 65
↪  38 64 63 36 62 63 37 37 61 36 30 33 36 33 66 66 66 30 63 33 37 35 38 39 65 31 62 36 31 32
↪  37 63 37 61 35 32 65 34 37 66 61 38 63 35 63 30 65 33 34 30 34 31 38 34 38 63 65 34 39 35
↪  30 61 37 34 34 63 65 39 31 66 36 62 33 61 35 30 38 66 66 64 31 63 38 65 22 2c 20 22 64 61
↪  74 61 5f 69 64 22 3a 20 30 7d 39 b3 59 ab
SJP: RX <- status: 0, service_id: 3, data_id: 0, data:
↪  "u'd5d1672dedcccda1026a7f34ab5ae8bec00e68f8583e8dc6bc77a60363fff0c37589e1b6127c7a52e47fa8
↪  c5c0e34041848ce4950a744ce91f6b3a508ffd1c8e'"
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 3.6.9 (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:
↪  "'ea1821be346b5df1cfdbfa5e7d6fe641161aafaf5ca1290b63691e712e74035b'"
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
↪  61 31 38 32 31 62 65 33 34 36 62 35 64 66 31 63 66 64 62 66 61 35 65 37 64 36 66 65 36 34
↪  31 31 36 31 61 61 66 61 66 35 63 61 31 32 39 30 62 36 33 36 39 31 65 37 31 32 65 37 34 30
↪  33 35 62 22 7d 29 51 80 d2
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 61 31 38 32 31 62 65 33 34 36 62 35 64 66 31 63 66 64 62 66 61 35 65 37 64 36 66 65 36
↪  34 31 31 36 31 61 61 66 61 66 35 63 61 31 32 39 30 62 36 33 36 39 31 65 37 31 32 65 37 34
↪  30 33 35 62 22 7d 29 51 80 d2
SJP: RX <- status: 0, service_id: 2, data_id: 0, data:
↪  "'ea1821be346b5df1cfdbfa5e7d6fe641161aafaf5ca1290b63691e712e74035b'"
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:
↪  "'fc81968a3390058acabf0fc03e0da12b20f2d0371bf5f84499ef9c592ab00a81131b51f9f78c75fa48a367c
↪  289ca9fd15239369ae2fdd76290cf22b17fe3f2f0'"
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 66
↪  63 38 31 39 36 38 61 33 33 39 30 30 35 38 61 63 61 62 66 30 66 63 30 33 65 30 64 61 31 32
↪  62 32 30 66 32 64 30 33 37 31 62 66 35 66 38 34 34 39 39 65 66 39 63 35 39 32 61 62 30 30
↪  61 38 31 31 33 31 62 35 31 66 39 66 37 38 63 37 35 66 61 34 38 61 33 36 37 63 32 38 39 63
↪  61 39 66 64 31 35 32 33 39 33 36 39 61 65 32 66 64 64 37 36 32 39 30 63 66 32 32 62 31 37
↪  66 65 33 66 32 66 30 22 7d 18 00 f1 87
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
↪  66 63 38 31 39 36 38 61 33 33 39 30 30 35 38 61 63 61 62 66 30 66 63 30 33 65 30 64 61 31
↪  32 62 32 30 66 32 64 30 33 37 31 62 66 35 66 38 34 34 39 39 65 66 39 63 35 39 32 61 62 30
↪  30 61 38 31 31 33 31 62 35 31 66 39 66 37 38 63 37 35 66 61 34 38 61 33 36 37 63 32 38 39
↪  63 61 39 66 64 31 35 32 33 39 33 36 39 61 65 32 66 64 64 37 36 32 39 30 63 66 32 32 62 31
↪  37 66 65 33 66 32 66 30 22 7d 18 00 f1 87
SJP: RX <- status: 0, service_id: 3, data_id: 0, data:
↪  "'fc81968a3390058acabf0fc03e0da12b20f2d0371bf5f84499ef9c592ab00a81131b51f9f78c75fa48a367c
↪  289ca9fd15239369ae2fdd76290cf22b17fe3f2f0'"
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 0.2016153335571289 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): 0.2016153335571289 (<class 'float'>)
Expectation (Timeout for authentification): 0.2 <= result <= 0.22000000000000003
```

Success	Timeout for authentification is correct (Content 0.5011227130889893 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): 0.5011227130889893 (<class 'float'>)
Expectation (Timeout for authentification): 0.5 <= result <= 0.55
```

Success	Timeout for send method is correct (Content 0.20057988166809082 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): 0.20057988166809082 (<class 'float'>)
Expectation (Timeout for send method): 0.2 <= result <= 0.22000000000000003
```

Success	Timeout for send method is correct (Content 0.5010545253753662 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): 0.5010545253753662 (<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:
↪  "'205036432263afdaefea69b4b21e10d2d5bbe93328b62442c834b9de746bd22f'"
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 32
↪  30 35 30 33 36 34 33 32 32 36 33 61 66 64 61 65 66 65 61 36 39 62 34 62 32 31 65 31 30 64
↪  32 64 35 62 62 65 39 33 33 32 38 62 36 32 34 34 32 63 38 33 34 62 39 64 65 37 34 36 62 64
↪  32 32 66 22 7d 98 1d 5c a6
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
↪  32 30 35 30 33 36 34 33 32 32 36 33 61 66 64 61 65 66 65 61 36 39 62 34 62 32 31 65 31 30
↪  64 32 64 35 62 62 65 39 33 33 32 38 62 36 32 34 34 32 63 38 33 34 62 39 64 65 37 34 36 62
↪  64 32 32 66 22 7d 98 1d 5c a6
SJP: RX <- status: 0, service_id: 2, data_id: 0, data:
↪  "'205036432263afdaefea69b4b21e10d2d5bbe93328b62442c834b9de746bd22f'"
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:
↪  "'d200de4b408079e1eb1d15dfd22fda38360433f1d1b51f9f45cea3c5a591052445391f87ca7b96216f77505
↪  7ff0cda065941b6e9227f32d9e328cbf0170e4674'"
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 64
↪  32 30 30 64 65 34 62 34 30 38 30 37 39 65 31 65 62 31 64 31 35 64 66 64 32 32 66 64 61 33
↪  38 33 36 30 34 33 33 66 31 64 31 62 35 31 66 39 66 34 35 63 65 61 33 63 35 61 35 39 31 30
↪  35 32 34 34 35 33 39 31 66 38 37 63 61 37 62 39 36 32 31 36 66 37 37 35 30 35 37 66 66 30
↪  63 64 61 30 36 35 39 34 31 62 36 65 39 32 32 37 66 33 32 64 39 65 33 32 38 63 62 66 30 31
↪  37 30 65 34 36 37 34 22 7d 5f 78 04 93
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
↪  64 32 30 30 64 65 34 62 34 30 38 30 37 39 65 31 65 62 31 64 31 35 64 66 64 32 32 66 64 61
↪  33 38 33 36 30 34 33 33 66 31 64 31 62 35 31 66 39 66 34 35 63 65 61 33 63 35 61 35 39 31
↪  30 35 32 34 34 35 33 39 31 66 38 37 63 61 37 62 39 36 32 31 36 66 37 37 35 30 35 37 66 66
↪  30 63 64 61 30 36 35 39 34 31 62 36 65 39 32 32 37 66 33 32 64 39 65 33 32 38 63 62 66 30
↪  31 37 30 65 34 36 37 34 22 7d 5f 78 04 93
SJP: RX <- status: 0, service_id: 3, data_id: 0, data:
↪  "'d200de4b408079e1eb1d15dfd22fda38360433f1d1b51f9f45cea3c5a591052445391f87ca7b96216f77505
↪  7ff0cda065941b6e9227f32d9e328cbf0170e4674'"
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 100.0%
The branch coverage for socket_protocol was 100.0%

C.1.1 socket_protocol.__init__.py

The line coverage for socket_protocol.__init__.py was 100.0%
The branch coverage for socket_protocol.__init__.py was 100.0%

```
#!/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


logger_name = 'SOCKET_PROTOCOL'
logger = logging.getLogger(logger_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


```

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

```
111     """
112     :param comm_instance: a communication instance supportin at least these functions: :func:`
        register_callback`, :func:`register_disconnect_callback`, :func:`send`.
113     :type comm_instance: instance
114     :param secret: A secret (e.g. created by ``binascii.hexlify(os.urandom(24))``).
115     :type secret: str
116
117     This communication protocol supports to transfer a Service-ID, Data-ID and Data. The
        transmitted data is shorter than :class:`pure_json_protocol`.
118
119     .. note::
120         This class is here for compatibility reasons. Usage of :class:`pure_json_protocol` is
        recommended.
121
122     **Example:**
123
124     Server:
125
126     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__struct_json_protocol_server.py
127
128     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__struct_json_protocol_server.log
129
130
131     Client:
132
133     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__struct_json_protocol_client.py
134
135     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__struct_json_protocol_client.log
136     """
137     LOG_PREFIX = 'SJP:'
138
139     SID_AUTH_SEED_REQUEST = 1
140     SID_AUTH_KEY_REQUEST = 2
141     SID_AUTH_KEY_CHECK_REQUEST = 3
142     SID_AUTH_KEY_CHECK_RESPONSE = 4
143     SID_READ_REQUEST = 10
144     SID_READ_RESPONSE = 11
145     SID_WRITE_REQUEST = 20
146     SID_WRITE_RESPONSE = 21
147     SID_EXECUTE_REQUEST = 30
148     SID_EXECUTE_RESPONSE = 31
149
150     SID_RESPONSE_DICT = {SID_AUTH_SEED_REQUEST: SID_AUTH_KEY_REQUEST,
151                          SID_AUTH_KEY_REQUEST: SID_AUTH_KEY_CHECK_REQUEST,
152                          SID_AUTH_KEY_CHECK_REQUEST: SID_AUTH_KEY_CHECK_RESPONSE,
153                          SID_READ_REQUEST: SID_READ_RESPONSE,
154                          SID_WRITE_REQUEST: SID_WRITE_RESPONSE,
155                          SID_EXECUTE_REQUEST: SID_EXECUTE_RESPONSE}
156
157     SID_AUTH_LIST = [SID_AUTH_SEED_REQUEST, SID_AUTH_KEY_REQUEST, SID_AUTH_KEY_CHECK_REQUEST,
        SID_AUTH_KEY_CHECK_RESPONSE]
158
159     STATUS_OKAY = 0
160     STATUS_BUFFERING_UNHANDLED_REQUEST = 1
161     STATUS_AUTH_REQUIRED = 2
162     STATUS_SERVICE_OR_DATA_UNKNOWN = 3
163     STATUS_CHECKSUM_ERROR = 4
164     STATUS_OPERATION_NOT_PERMITTED = 5
165     STATUS_NAMES = {STATUS_OKAY: 'Okay',
```

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

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

```
279                 else:
280                     try:
281                         logger.debug("%s Executing callback %s to process received data", self.
    LOG_PREFIX, callback.__name__)
282                         status, data = callback(msg)
283                     except TypeError:
284                         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())))
285 
286     def __buffer_received_data__(self, msg):
287         if not msg.get_service_id() in self.__msg_buffer__:
288             self.__msg_buffer__[msg.get_service_id()] = {}
289         if not msg.get_data_id() in self.__msg_buffer__[msg.get_service_id()]:
290             self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()] = []
291         self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()].append(msg)
292         logger.debug("%s Message data is stored in buffer and is now ready to be retrieved by
    receive method", self.LOG_PREFIX)
293 
294     def __clean_receive_buffer__(self):
295         logger.debug("%s Cleaning up receive-buffer", self.LOG_PREFIX)
296         self.__msg_buffer__ = {}
297 
298     def receive(self, service_id, data_id, timeout=1):
299         data = None
300         cnt = 0
301         while data is None and cnt < timeout * 10:
302             try:
303                 data = self.__msg_buffer__.get(service_id, {}).get(data_id, []).pop(0)
304             except IndexError:
305                 data = None
306             cnt += 1
307             time.sleep(0.1)
308         if data is None and cnt >= timeout * 10:
309             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))
310         return data
311 
312     def __mk_msg__(self, status, service_id, data_id, data):
313         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})
314 
315     def send(self, service_id, data_id, data, status=STATUS_OKAY, timeout=2, log_lvl=logging.INFO
    ):
316         """
317         :param service_id: The Service-ID for the message. See class definitions starting with ``
    SERVICE_``.
318         :type service_id: int
319         :param data_id: The Data-ID for the message.
320         :type data_id: int
321         :param data: The data to be transfered. The data needs to be json compatible.
322         :type data: str
323         :param status: The Status for the message. All requests should have ``STATUS_OKAY``.
324         :type status: int
325         :param timeout: The timeout for sending data (e.g. time to establish new connection).
326         :type timeout: float
327         :param rx_log_lvl: The log level to log outgoing TX-data
328         :type rx_log_lvl: int
329         :return: True if data had been sent, otherwise False.
330         :rtype: bool
331 
332         This methods sends out a message with the given content.
333         """
```

```
334         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))
335         return self.__comm_inst__.send(self.__build_frame__(service_id, data_id, data, status),
    timeout=timeout, log_lvl=logging.DEBUG)
336 
337     def register_callback(self, service_id, data_id, callback):
338         """
339         :param service_id: The Service-ID for the message. See class definitions starting with ``
    SID_``.
340         :type service_id: int
341         :param data_id: The Data-ID for the message.
342         :type data_id: int
343         :returns: True, if registration was successfull; False, if registration failed (e.g.
    existance of a callback for this configuration)
344         :rtype: bool
345 
346         This method registers a callback for the given parameters. Givin ``None`` means, that all
     Service-IDs or all Data-IDs are used.
347         If a message hitting these parameters has been received, the callback will be executed.
348 
349         .. note:: The :func:`callback` is priorised in the following order:
350 
351             * Callbacks with defined Service-ID and Data-ID.
352             * Callbacks with a defined Data-ID.
353             * Callbacks with a defined Service-ID.
354             * Unspecific Callbacks
355 
356         .. note:: The :func:`callback` is executed with these arguments:
357 
358             :param msg: A :class:`dict` containing all message information.
359             :returns: status (see class definition starting with ``STATUS_``), response_data (
    JSON compatible object)
360         """
361         self.__callbacks__.add(service_id, data_id, callback)
362 
363     def authentificate(self, timeout=2):
364         """
365         :param timeout: The timeout for the authentification (requesting seed, sending key and
    getting authentification_feedback).
366         :type timeout: float
367         :returns: True, if authentification was successfull; False, if not.
368         :rtype: bool
369 
370         This method authetificates the client at the server.
371 
372         .. note:: An authentification will only processed, if a secret had been given on
    initialisation.
373 
374         .. note:: Client and Server needs to use the same secret.
375         """
376         if self.__secret__ is not None:
377             self.__authentification_state__ = self.AUTH_STATE_SEED_REQUESTED
378             logger.info("%s Requesting seed for authentification", self.LOG_PREFIX)
379             self.send(self.SID_AUTH_SEED_REQUEST, 0, None)
380             cnt = 0
381             while cnt < timeout * 10:
382                 time.sleep(0.1)
383                 if self.__authentification_state__ == self.AUTH_STATE_TRUSTED_CLIENT:
384                     return True
385                 elif self.__authentification_state__ == self.AUTH_STATE_UNKNOWN_CLIENT:
386                     break
387                 cnt += 1
388         return False
```

```
389
390     def check_authentification_state(self):
391         """
392         :return: True, if authentification state is okay, otherwise False
393         :rtype: bool
394         """
395         return self.__authentification_state__ == self.AUTH_STATE_TRUSTED_CLIENT
396
397     def __authentificate_salt_and_hash__(self, seed):
398         if sys.version_info >= (3, 0):
399             return hashlib.sha512(bytes(seed, 'utf-8') + self.__secret__).hexdigest()
400         else:
401             return hashlib.sha512(seed.encode('utf-8') + self.__secret__.encode('utf-8')).
    hexdigest()
402
403     def __authentificate_create_seed__(self, msg):
404         logger.info("%s Got seed request, sending seed for authentification", self.LOG_PREFIX)
405         self.__authentification_state__ = self.AUTH_STATE_SEED_TRANSFERRED
406         if sys.version_info >= (3, 0):
407             self.__seed__ = binascii.hexlify(os.urandom(32)).decode('utf-8')
408         else:
409             self.__seed__ = binascii.hexlify(os.urandom(32))
410         return self.STATUS_OKAY, self.__seed__
411
412     def __authentificate_create_key__(self, msg):
413         logger.info("%s Got seed, sending key for authentification", self.LOG_PREFIX)
414         self.__authentification_state__ = self.AUTH_STATE_KEY_TRANSFERRED
415         seed = msg.get_data()
416         key = self.__authentificate_salt_and_hash__(seed)
417         return self.STATUS_OKAY, key
418
419     def __authentificate_check_key__(self, msg):
420         key = msg.get_data()
421         if key == self.__authentificate_salt_and_hash__(self.__seed__):
422             self.__authentification_state__ = self.AUTH_STATE_TRUSTED_CLIENT
423             logger.info("%s Got correct key, sending positive authentification feedback", self.
    LOG_PREFIX)
424             return self.STATUS_OKAY, True
425         else:
426             self.__authentification_state__ = self.AUTH_STATE_UNKNOWN_CLIENT
427             logger.info("%s Got incorrect key, sending negative authentification feedback", self.
    LOG_PREFIX)
428             return self.STATUS_OKAY, False
429
430     def __authentificate_process_feedback__(self, msg):
431         feedback = msg.get_data()
432         if feedback:
433             self.__authentification_state__ = self.AUTH_STATE_TRUSTED_CLIENT
434             logger.info("%s Got positive authentification feedback", self.LOG_PREFIX)
435         else:
436             self.__authentification_state__ = self.AUTH_STATE_UNKNOWN_CLIENT
437             logger.warning("%s Got negative authentification feedback", self.LOG_PREFIX)
438         return self.STATUS_OKAY, None
439
440
441 class pure_json_protocol(struct_json_protocol):
```

```
442     """
443     :param comm_instance: a communication instance supportin at least these functions: :func:`
        register_callback`, :func:`register_disconnect_callback`, :func:`send`.
444     :type comm_instance: instance
445     :param secret: A secret (e.g. created by ``binascii.hexlify(os.urandom(24))``).
446     :type secret: str
447
448     This communication protocol supports to transfer a Service-ID, Data-ID and Data.
449
450     **Example:**
451
452     Server:
453
454     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__pure_json_protocol_server.py
455
456     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__pure_json_protocol_server.log
457
458
459     Client:
460
461     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__pure_json_protocol_client.py
462
463     .. literalinclude:: ../../socket_protocol/_examples_/
        socket_protocol__pure_json_protocol_client.log
464     """
465     def __init__(self, comm_instance, secret=None):
466         struct_json_protocol.__init__(self, comm_instance, secret)
467
468     def __build_frame__(self, service_id, data_id, data, status=struct_json_protocol.STATUS_OKAY)
        :
469         data_frame = json.dumps(self.__mk_msg__(status, service_id, data_id, data))
470         if sys.version_info >= (3, 0):
471             data_frame = bytes(data_frame, 'utf-8')
472         checksum = self.__calc_chksum__(data_frame)
473         return data_frame + checksum
474
475     def __analyse_frame__(self, frame):
476         if sys.version_info >= (3, 0):
477             return data_storage(json.loads(frame[:-4].decode('utf-8')))
478         else:
479             return data_storage(json.loads(frame[:-4]))
480
481     def __calc_chksum__(self, raw_data):
482         return struct.pack('>I', binascii.crc32(raw_data) & 0xffffffff)
483
484     def __check_frame_checksum__(self, frame):
485         return self.__calc_chksum__(frame[:-4]) == frame[-4:]
```