

Unittest for report

January 2, 2020

Contents

1	Test Information	3
1.1	Test Candidate Information	3
1.2	Unittest Information	3
1.3	Test System Information	3
2	Statistic	3
2.1	Test-Statistic for testrun with python 2.7.17 (final)	3
2.2	Test-Statistic for testrun with python 3.6.9 (final)	4
2.3	Coverage Statistic	4
3	Tested Requirements	5
3.1	General Information	5
3.2	collectingHandler	5
3.2.1	Store log records (collectingHandler)	5
3.2.2	String representation (collectingHandler)	6
3.3	collectingRingHandler	7
3.3.1	Store log records (collectingRingHandler)	7
3.3.2	String representation (collectingRingHandler)	8
3.3.3	Number of stored logs in the RingHandler	8
A	Trace for testrun with python 2.7.17 (final)	10
A.1	Tests with status Info (5)	10
A.1.1	Store log records (collectingHandler)	10
A.1.2	String representation (collectingHandler)	11
A.1.3	Store log records (collectingRingHandler)	13
A.1.4	String representation (collectingRingHandler)	14
A.1.5	Number of stored logs in the RingHandler	15
B	Trace for testrun with python 3.6.9 (final)	16
B.1	Tests with status Info (5)	16
B.1.1	Store log records (collectingHandler)	16
B.1.2	String representation (collectingHandler)	18
B.1.3	Store log records (collectingRingHandler)	19
B.1.4	String representation (collectingRingHandler)	20
B.1.5	Number of stored logs in the RingHandler	21

C Test-Coverage	22
C.1 report	22
C.1.1 report.__init__.py	23

1 Test Information

1.1 Test Candidate Information

The Module `report` is designed to help with python logging and to support some handlers for logging to memory. For more Information read the sphinx documentation.

Library Information

Name	report
State	Released
Supported Interpreters	python2, python3
Version	1c229da8210becc90fd74efd75907007

Dependencies

1.2 Unittest Information

Unittest Information

Version	2c03d3eba161a9fb0dbf0594fbda3965
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	5.0.0-37-generic (#40 18.04.1-Ubuntu SMP Thu Nov 14 12:06:39 UTC 2019)
Machine	x86_64
Path	/user_data/data/dirk/prj/modules/report/unittest
System	Linux
Username	dirk

2 Statistic

2.1 Test-Statistic for testrun with python 2.7.17 (final)

Number of tests	5
Number of successfull tests	5
Number of possibly failed tests	0
Number of failed tests	0

Executionlevel	Full Test (all defined tests)
Time consumption	0.010s

2.2 Test-Statistic for testrun with python 3.6.9 (final)

Number of tests	5
Number of successfull tests	5
Number of possibly failed tests	0
Number of failed tests	0

Executionlevel	Full Test (all defined tests)
Time consumption	0.034s

2.3 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
report	89.9%	67.7%
report.__init__.py	89.9%	

3 Tested Requirements

3.1 General Information

Many Methods and Classes in this Module are used for the unittest itself. Others are configuring python logging, which is also used for the unittest itself. Therefore, the unittest for this Module is limited. Also the coverage information is not only reached by the testcases, cause the Module is used by the unittest itself.

3.2 collectingHandler

3.2.1 Store log records (collectingHandler)

Description

Description 1.

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/modules/report/unittest/src/tests/_init_.py (19)
Start-Time:	2020-01-02 14:18:00,610
Finished-Time:	2020-01-02 14:18:00,612
Time-Consumption	0.002s

Testsummary:

Info	Running logger test sequence.
Success	Length of collected logs is correct (Content 7 and Type is <type 'int'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 10, 'test_helpers.py', 1] and Type is <type 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 2] and Type is <type 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 30, 'test_helpers.py', 3] and Type is <type 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 4] and Type is <type 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 50, 'test_helpers.py', 5] and Type is <type 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 6] and Type is <type 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 7] and Type is <type 'list'>).

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/modules/report/unittest/src/tests/_init_.py (19)

Start-Time: 2020-01-02 14:18:01,250
 Finished-Time: 2020-01-02 14:18:01,255
 Time-Consumption 0.005s

Testsummary:

Info Running logger test sequence.
Success Length of collected logs is correct (Content 7 and Type is <class 'int'>).
Success Logged information is correct (Content ['Log entry number %d with level %s.', 10, 'test_helpers.py', 1] and Type is <class 'list'>).
Success Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 2] and Type is <class 'list'>).
Success Logged information is correct (Content ['Log entry number %d with level %s.', 30, 'test_helpers.py', 3] and Type is <class 'list'>).
Success Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 4] and Type is <class 'list'>).
Success Logged information is correct (Content ['Log entry number %d with level %s.', 50, 'test_helpers.py', 5] and Type is <class 'list'>).
Success Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 6] and Type is <class 'list'>).
Success Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 7] and Type is <class 'list'>).

3.2.2 String representation (collectingHandler)**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/modules/report/unittest/src/tests/___init___py (20)
 Start-Time: 2020-01-02 14:18:00,612
 Finished-Time: 2020-01-02 14:18:00,614
 Time-Consumption 0.002s

Testsummary:

Info Running logger test sequence.
Success Indexlist of log entries in stringrepresentation: Values and number of submitted values is correct. See detailed log for more information.

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/modules/report/unittest/src/tests/___init___py (20)
 Start-Time: 2020-01-02 14:18:01,255
 Finished-Time: 2020-01-02 14:18:01,264
 Time-Consumption 0.009s

Testsummary:

Info Running logger test sequence.

Success Indexlist of log entries in stringrepresentation: Values and number of submitted values is correct. See detailed log for more information.

3.3 collectingRingHandler

3.3.1 Store log records (collectingRingHandler)

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/modules/report/unittest/src/tests/_init_.py (21)
Start-Time:	2020-01-02 14:18:00,614
Finished-Time:	2020-01-02 14:18:00,616
Time-Consumption	0.002s

Testsummary:

Info	Running logger test sequence.
Success	Length of collected logs is correct (Content 5 and Type is <type 'int'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 30, 'test_helpers.py', 3] and Type is <type 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 4] and Type is <type 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 50, 'test_helpers.py', 5] and Type is <type 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 6] and Type is <type 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 7] and Type is <type 'list'>).

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/modules/report/unittest/src/tests/_init_.py (21)
Start-Time:	2020-01-02 14:18:01,267
Finished-Time:	2020-01-02 14:18:01,272
Time-Consumption	0.005s

Testsummary:

Info	Running logger test sequence.
Success	Length of collected logs is correct (Content 5 and Type is <class 'int'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 30, 'test_helpers.py', 3] and Type is <class 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 4] and Type is <class 'list'>).
Success	Logged information is correct (Content ['Log entry number %d with level %s.', 50, 'test_helpers.py', 5] and Type is <class 'list'>).

Success Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 6] and Type is <class 'list'>).

Success Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 7] and Type is <class 'list'>).

3.3.2 String representation (collectingRingHandler)

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/modules/report/unittest/src/tests/...init...py (22)
Start-Time:	2020-01-02 14:18:00,616
Finished-Time:	2020-01-02 14:18:00,618
Time-Consumption	0.002s

Testsummary:

Info Running logger test sequence.

Success Indexlist of log entries in stringrepresentation: Values and number of submitted values is correct. See detailed log for more information.

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/modules/report/unittest/src/tests/...init...py (22)
Start-Time:	2020-01-02 14:18:01,272
Finished-Time:	2020-01-02 14:18:01,281
Time-Consumption	0.009s

Testsummary:

Info Running logger test sequence.

Success Indexlist of log entries in stringrepresentation: Values and number of submitted values is correct. See detailed log for more information.

3.3.3 Number of stored logs in the RingHandler

Description

The number of stored log-records shall be given on initialisation or reinitialisation.

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/modules/report/unittest/src/tests/...init...py (23)
Start-Time:	2020-01-02 14:18:00,619

Unittest for report

Finished-Time: 2020-01-02 14:18:00,621
Time-Consumption 0.002s

Testsummary:

Info	Running logger test sequence.
Success	Length of collectingRingHandler is correct (Content 5 and Type is <type 'int'>).
Success	Length of collectingRingHandler after reinitialisation is correct (Content 3 and Type is <type 'int'>).
Success	Log text is correct (Content 'Log entry number 5 with level CRITICAL.' and Type is <type 'str'>).
Success	Log text is correct (Content 'Log entry number 6 with level INFO.' and Type is <type 'str'>).
Success	Log text is correct (Content 'Log entry number 7 with level ERROR.' and Type is <type 'str'>).

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/modules/report/unittest/src/tests/___init___py (23)
Start-Time:	2020-01-02 14:18:01,282
Finished-Time:	2020-01-02 14:18:01,287
Time-Consumption	0.005s

Testsummary:

Info	Running logger test sequence.
Success	Length of collectingRingHandler is correct (Content 5 and Type is <class 'int'>).
Success	Length of collectingRingHandler after reinitialisation is correct (Content 3 and Type is <class 'int'>).
Success	Log text is correct (Content 'Log entry number 5 with level CRITICAL.' and Type is <class 'str'>).
Success	Log text is correct (Content 'Log entry number 6 with level INFO.' and Type is <class 'str'>).
Success	Log text is correct (Content 'Log entry number 7 with level ERROR.' and Type is <class 'str'>).

A Trace for testrun with python 2.7.17 (final)

A.1 Tests with status Info (5)

A.1.1 Store log records (collectingHandler)

Description

Description 1.

Testresult

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

Info Running logger test sequence.

Configuring collecting logger

Passing "Log entry number 1 with level DEBUG." to collectingHandler.

Passing "Log entry number 2 with level INFO." to collectingHandler.

Passing "Log entry number 3 with level WARNING." to collectingHandler.

Passing "Log entry number 4 with level ERROR." to collectingHandler.

Passing "Log entry number 5 with level CRITICAL." to collectingHandler.

Passing "Log entry number 6 with level INFO." to collectingHandler.

Passing "Log entry number 7 with level ERROR." to collectingHandler.

Success Length of collected logs is correct (Content 7 and Type is <type 'int'>).

Result (Length of collected logs): 7 (<type 'int'>)

Expectation (Length of collected logs): result = 7 (<type 'int'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 10, 'test_helpers.py', 1] and Type is <type 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 10, 'test_helpers.py', 1
↪] (<type 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 10,
↪ 'test_helpers.py', 1] (<type 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 2] and Type is <type 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 20, 'test_helpers.py', 2
↪] (<type 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 20,
↪ 'test_helpers.py', 2] (<type 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 30, 'test_helpers.py', 3] and Type is <type 'list'>).

```
Result (Logged information): [ 'Log entry number %d with level %s.', 30, 'test_helpers.py', 3
↵ ] (<type 'list'>)
```

```
Expectation (Logged information): result = [ 'Log entry number %d with level %s.', 30,
↵ 'test_helpers.py', 3 ] (<type 'list'>)
```

Success Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 4] and Type is <type 'list'>).

```
Result (Logged information): [ 'Log entry number %d with level %s.', 40, 'test_helpers.py', 4
↵ ] (<type 'list'>)
```

```
Expectation (Logged information): result = [ 'Log entry number %d with level %s.', 40,
↵ 'test_helpers.py', 4 ] (<type 'list'>)
```

Success Logged information is correct (Content ['Log entry number %d with level %s.', 50, 'test_helpers.py', 5] and Type is <type 'list'>).

```
Result (Logged information): [ 'Log entry number %d with level %s.', 50, 'test_helpers.py', 5
↵ ] (<type 'list'>)
```

```
Expectation (Logged information): result = [ 'Log entry number %d with level %s.', 50,
↵ 'test_helpers.py', 5 ] (<type 'list'>)
```

Success Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 6] and Type is <type 'list'>).

```
Result (Logged information): [ 'Log entry number %d with level %s.', 20, 'test_helpers.py', 6
↵ ] (<type 'list'>)
```

```
Expectation (Logged information): result = [ 'Log entry number %d with level %s.', 20,
↵ 'test_helpers.py', 6 ] (<type 'list'>)
```

Success Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 7] and Type is <type 'list'>).

```
Result (Logged information): [ 'Log entry number %d with level %s.', 40, 'test_helpers.py', 7
↵ ] (<type 'list'>)
```

```
Expectation (Logged information): result = [ 'Log entry number %d with level %s.', 40,
↵ 'test_helpers.py', 7 ] (<type 'list'>)
```

A.1.2 String representation (collectingHandler)

Testresult

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

Info Running logger test sequence.

Configuring collecting logger

Passing "Log entry number 1 with level DEBUG." to collectingHandler.

Passing "Log entry number 2 with level INFO." to collectingHandler.

Passing "Log entry number 3 with level WARNING." to collectingHandler.

Passing "Log entry number 4 with level ERROR." to collectingHandler.

Passing "Log entry number 5 with level CRITICAL." to collectingHandler.

Passing "Log entry number 6 with level INFO." to collectingHandler.

Passing "Log entry number 7 with level ERROR." to collectingHandler.

Success Indexlist of log entries in stringrepresentation: Values and number of submitted values is correct. See detailed log for more information.

Result (Indexlist of log entries in stringrepresentation): [35, 107, 178, 252, 324, 399, 470
↪] (<type 'list'>)

Expectation (Indexlist of log entries in stringrepresentation): result = [35, 107, 178, 252,
↪ 324, 399, 470] (<type 'list'>)

Result (Submitted value number 1): 35 (<type 'int'>)

Expectation (Submitted value number 1): result = 35 (<type 'int'>)

Submitted value number 1 is correct (Content 35 and Type is <type 'int'>).

Result (Submitted value number 2): 107 (<type 'int'>)

Expectation (Submitted value number 2): result = 107 (<type 'int'>)

Submitted value number 2 is correct (Content 107 and Type is <type 'int'>).

Result (Submitted value number 3): 178 (<type 'int'>)

Expectation (Submitted value number 3): result = 178 (<type 'int'>)

Submitted value number 3 is correct (Content 178 and Type is <type 'int'>).

Result (Submitted value number 4): 252 (<type 'int'>)

Expectation (Submitted value number 4): result = 252 (<type 'int'>)

Submitted value number 4 is correct (Content 252 and Type is <type 'int'>).

Result (Submitted value number 5): 324 (<type 'int'>)

Expectation (Submitted value number 5): result = 324 (<type 'int'>)

Submitted value number 5 is correct (Content 324 and Type is <type 'int'>).

Result (Submitted value number 6): 399 (<type 'int'>)

Expectation (Submitted value number 6): result = 399 (<type 'int'>)

Submitted value number 6 is correct (Content 399 and Type is <type 'int'>).

Result (Submitted value number 7): 470 (<type 'int'>)

Expectation (Submitted value number 7): result = 470 (<type 'int'>)

Submitted value number 7 is correct (Content 470 and Type is <type 'int'>).

A.1.3 Store log records (collectingRingHandler)

Testresult

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

Info Running logger test sequence.

Configuring collecting logger

Passing "Log entry number 1 with level DEBUG." to collectingRingHandler.

Passing "Log entry number 2 with level INFO." to collectingRingHandler.

Passing "Log entry number 3 with level WARNING." to collectingRingHandler.

Passing "Log entry number 4 with level ERROR." to collectingRingHandler.

Passing "Log entry number 5 with level CRITICAL." to collectingRingHandler.

Passing "Log entry number 6 with level INFO." to collectingRingHandler.

Passing "Log entry number 7 with level ERROR." to collectingRingHandler.

Success Length of collected logs is correct (Content 5 and Type is <type 'int'>).

Result (Length of collected logs): 5 (<type 'int'>)

Expectation (Length of collected logs): result = 5 (<type 'int'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 30, 'test_helpers.py', 3] and Type is <type 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 30, 'test_helpers.py', 3
↪] (<type 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 30,
↪ 'test_helpers.py', 3] (<type 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 4] and Type is <type 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 40, 'test_helpers.py', 4
↪] (<type 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 40,
↪ 'test_helpers.py', 4] (<type 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 50, 'test_helpers.py', 5] and Type is <type 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 50, 'test_helpers.py', 5
↪] (<type 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 50,
↪ 'test_helpers.py', 5] (<type 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 6] and Type is <type 'list'>).

```
Result (Logged information): [ 'Log entry number %d with level %s.', 20, 'test_helpers.py', 6  
↪ ] (<type 'list'>)
```

```
Expectation (Logged information): result = [ 'Log entry number %d with level %s.', 20,  
↪ 'test_helpers.py', 6 ] (<type 'list'>)
```

Success Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 7] and Type is <type 'list'>).

```
Result (Logged information): [ 'Log entry number %d with level %s.', 40, 'test_helpers.py', 7  
↪ ] (<type 'list'>)
```

```
Expectation (Logged information): result = [ 'Log entry number %d with level %s.', 40,  
↪ 'test_helpers.py', 7 ] (<type 'list'>)
```

A.1.4 String representation (collectingRingHandler)

Testresult

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

Info Running logger test sequence.

Configuring collecting logger

Passing "Log entry number 1 with level DEBUG." to collectingRingHandler.

Passing "Log entry number 2 with level INFO." to collectingRingHandler.

Passing "Log entry number 3 with level WARNING." to collectingRingHandler.

Passing "Log entry number 4 with level ERROR." to collectingRingHandler.

Passing "Log entry number 5 with level CRITICAL." to collectingRingHandler.

Passing "Log entry number 6 with level INFO." to collectingRingHandler.

Passing "Log entry number 7 with level ERROR." to collectingRingHandler.

Success Indexlist of log entries in stringrepresentation: Values and number of submitted values is correct. See detailed log for more information.

```

Result (Indexlist of log entries in stringrepresentation): [ 35, 109, 181, 256, 327 ] (<type 'list'>)
↳ 'list'>)
Expectation (Indexlist of log entries in stringrepresentation): result = [ 35, 109, 181, 256,
↳ 327 ] (<type 'list'>)
Result (Submitted value number 1): 35 (<type 'int'>)
Expectation (Submitted value number 1): result = 35 (<type 'int'>)
Submitted value number 1 is correct (Content 35 and Type is <type 'int'>).
Result (Submitted value number 2): 109 (<type 'int'>)
Expectation (Submitted value number 2): result = 109 (<type 'int'>)
Submitted value number 2 is correct (Content 109 and Type is <type 'int'>).
Result (Submitted value number 3): 181 (<type 'int'>)
Expectation (Submitted value number 3): result = 181 (<type 'int'>)
Submitted value number 3 is correct (Content 181 and Type is <type 'int'>).
Result (Submitted value number 4): 256 (<type 'int'>)
Expectation (Submitted value number 4): result = 256 (<type 'int'>)
Submitted value number 4 is correct (Content 256 and Type is <type 'int'>).
Result (Submitted value number 5): 327 (<type 'int'>)
Expectation (Submitted value number 5): result = 327 (<type 'int'>)
Submitted value number 5 is correct (Content 327 and Type is <type 'int'>).

```

A.1.5 Number of stored logs in the RingHandler

Description

The number of stored log-records shall be given on initialisation or reinitialisation.

Testresult

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

Info Running logger test sequence.

Configuring collecting logger

Passing "Log entry number 1 with level DEBUG." to collectingRingHandler.

Passing "Log entry number 2 with level INFO." to collectingRingHandler.

Passing "Log entry number 3 with level WARNING." to collectingRingHandler.

Passing "Log entry number 4 with level ERROR." to collectingRingHandler.

Passing "Log entry number 5 with level CRITICAL." to collectingRingHandler.

Passing "Log entry number 6 with level INFO." to collectingRingHandler.

Passing "Log entry number 7 with level ERROR." to collectingRingHandler.

Success Length of collectingRingHandler is correct (Content 5 and Type is <type 'int'>).

Result (Length of collectingRingHandler): 5 (<type 'int'>)

Expectation (Length of collectingRingHandler): result = 5 (<type 'int'>)

Success Length of collectingRingHandler after reinitialisation is correct (Content 3 and Type is <type 'int'>).

```
Result (Length of collectingRingHandler after reinitialisation): 3 (<type 'int'>)
```

```
Expectation (Length of collectingRingHandler after reinitialisation): result = 3 (<type  
↪ 'int'>)
```

Success Log text is correct (Content 'Log entry number 5 with level CRITICAL.' and Type is <type 'str'>).

```
Result (Log text): 'Log entry number 5 with level CRITICAL.' (<type 'str'>)
```

```
Expectation (Log text): result = 'Log entry number 5 with level CRITICAL.' (<type 'str'>)
```

Success Log text is correct (Content 'Log entry number 6 with level INFO.' and Type is <type 'str'>).

```
Result (Log text): 'Log entry number 6 with level INFO.' (<type 'str'>)
```

```
Expectation (Log text): result = 'Log entry number 6 with level INFO.' (<type 'str'>)
```

Success Log text is correct (Content 'Log entry number 7 with level ERROR.' and Type is <type 'str'>).

```
Result (Log text): 'Log entry number 7 with level ERROR.' (<type 'str'>)
```

```
Expectation (Log text): result = 'Log entry number 7 with level ERROR.' (<type 'str'>)
```

B Trace for testrun with python 3.6.9 (final)

B.1 Tests with status Info (5)

B.1.1 Store log records (collectingHandler)

Description

Description 1.

Testresult

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

Info Running logger test sequence.

```
Configuring collecting logger
```

```
Passing "Log entry number 1 with level DEBUG." to collectingHandler.
```

```
Passing "Log entry number 2 with level INFO." to collectingHandler.
```

```
Passing "Log entry number 3 with level WARNING." to collectingHandler.
```

```
Passing "Log entry number 4 with level ERROR." to collectingHandler.
```

```
Passing "Log entry number 5 with level CRITICAL." to collectingHandler.
```

```
Passing "Log entry number 6 with level INFO." to collectingHandler.
```

```
Passing "Log entry number 7 with level ERROR." to collectingHandler.
```

Success Length of collected logs is correct (Content 7 and Type is <class 'int'>).

Result (Length of collected logs): 7 (<class 'int'>)

Expectation (Length of collected logs): result = 7 (<class 'int'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 10, 'test_helpers.py', 1] and Type is <class 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 10, 'test_helpers.py', 1
↪] (<class 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 10,
↪ 'test_helpers.py', 1] (<class 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 2] and Type is <class 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 20, 'test_helpers.py', 2
↪] (<class 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 20,
↪ 'test_helpers.py', 2] (<class 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 30, 'test_helpers.py', 3] and Type is <class 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 30, 'test_helpers.py', 3
↪] (<class 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 30,
↪ 'test_helpers.py', 3] (<class 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 4] and Type is <class 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 40, 'test_helpers.py', 4
↪] (<class 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 40,
↪ 'test_helpers.py', 4] (<class 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 50, 'test_helpers.py', 5] and Type is <class 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 50, 'test_helpers.py', 5
↪] (<class 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 50,
↪ 'test_helpers.py', 5] (<class 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 6] and Type is <class 'list'>).

```
Result (Logged information): [ 'Log entry number %d with level %s.', 20, 'test_helpers.py', 6  
↪ ] (<class 'list'>)
```

```
Expectation (Logged information): result = [ 'Log entry number %d with level %s.', 20,  
↪ 'test_helpers.py', 6 ] (<class 'list'>)
```

Success Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 7] and Type is <class 'list'>).

```
Result (Logged information): [ 'Log entry number %d with level %s.', 40, 'test_helpers.py', 7  
↪ ] (<class 'list'>)
```

```
Expectation (Logged information): result = [ 'Log entry number %d with level %s.', 40,  
↪ 'test_helpers.py', 7 ] (<class 'list'>)
```

B.1.2 String representation (collectingHandler)

Testresult

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

Info Running logger test sequence.

```
Configuring collecting logger
```

```
Passing "Log entry number 1 with level DEBUG." to collectingHandler.
```

```
Passing "Log entry number 2 with level INFO." to collectingHandler.
```

```
Passing "Log entry number 3 with level WARNING." to collectingHandler.
```

```
Passing "Log entry number 4 with level ERROR." to collectingHandler.
```

```
Passing "Log entry number 5 with level CRITICAL." to collectingHandler.
```

```
Passing "Log entry number 6 with level INFO." to collectingHandler.
```

```
Passing "Log entry number 7 with level ERROR." to collectingHandler.
```

Success Indexlist of log entries in stringrepresentation: Values and number of submitted values is correct. See detailed log for more information.

Result (Indexlist of log entries in stringrepresentation): [35, 107, 178, 252, 324, 399, 470
 ↪] (<class 'list'>)

Expectation (Indexlist of log entries in stringrepresentation): result = [35, 107, 178, 252,
 ↪ 324, 399, 470] (<class 'list'>)

Result (Submitted value number 1): 35 (<class 'int'>)

Expectation (Submitted value number 1): result = 35 (<class 'int'>)

Submitted value number 1 is correct (Content 35 and Type is <class 'int'>).

Result (Submitted value number 2): 107 (<class 'int'>)

Expectation (Submitted value number 2): result = 107 (<class 'int'>)

Submitted value number 2 is correct (Content 107 and Type is <class 'int'>).

Result (Submitted value number 3): 178 (<class 'int'>)

Expectation (Submitted value number 3): result = 178 (<class 'int'>)

Submitted value number 3 is correct (Content 178 and Type is <class 'int'>).

Result (Submitted value number 4): 252 (<class 'int'>)

Expectation (Submitted value number 4): result = 252 (<class 'int'>)

Submitted value number 4 is correct (Content 252 and Type is <class 'int'>).

Result (Submitted value number 5): 324 (<class 'int'>)

Expectation (Submitted value number 5): result = 324 (<class 'int'>)

Submitted value number 5 is correct (Content 324 and Type is <class 'int'>).

Result (Submitted value number 6): 399 (<class 'int'>)

Expectation (Submitted value number 6): result = 399 (<class 'int'>)

Submitted value number 6 is correct (Content 399 and Type is <class 'int'>).

Result (Submitted value number 7): 470 (<class 'int'>)

Expectation (Submitted value number 7): result = 470 (<class 'int'>)

Submitted value number 7 is correct (Content 470 and Type is <class 'int'>).

B.1.3 Store log records (collectingRingHandler)

Testresult

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

Info Running logger test sequence.

Configuring collecting logger

Passing "Log entry number 1 with level DEBUG." to collectingRingHandler.

Passing "Log entry number 2 with level INFO." to collectingRingHandler.

Passing "Log entry number 3 with level WARNING." to collectingRingHandler.

Passing "Log entry number 4 with level ERROR." to collectingRingHandler.

Passing "Log entry number 5 with level CRITICAL." to collectingRingHandler.

Passing "Log entry number 6 with level INFO." to collectingRingHandler.

Passing "Log entry number 7 with level ERROR." to collectingRingHandler.

Success Length of collected logs is correct (Content 5 and Type is <class 'int'>).

Result (Length of collected logs): 5 (<class 'int'>)

Expectation (Length of collected logs): result = 5 (<class 'int'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 30, 'test_helpers.py', 3] and Type is <class 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 30, 'test_helpers.py', 3
↪] (<class 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 30,
↪ 'test_helpers.py', 3] (<class 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 4] and Type is <class 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 40, 'test_helpers.py', 4
↪] (<class 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 40,
↪ 'test_helpers.py', 4] (<class 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 50, 'test_helpers.py', 5] and Type is <class 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 50, 'test_helpers.py', 5
↪] (<class 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 50,
↪ 'test_helpers.py', 5] (<class 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 20, 'test_helpers.py', 6] and Type is <class 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 20, 'test_helpers.py', 6
↪] (<class 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 20,
↪ 'test_helpers.py', 6] (<class 'list'>)

Success Logged information is correct (Content ['Log entry number %d with level %s.', 40, 'test_helpers.py', 7] and Type is <class 'list'>).

Result (Logged information): ['Log entry number %d with level %s.', 40, 'test_helpers.py', 7
↪] (<class 'list'>)

Expectation (Logged information): result = ['Log entry number %d with level %s.', 40,
↪ 'test_helpers.py', 7] (<class 'list'>)

B.1.4 String representation (collectingRingHandler)

Testresult

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

Info Running logger test sequence.

Configuring collecting logger

Passing "Log entry number 1 with level DEBUG." to collectingRingHandler.

Passing "Log entry number 2 with level INFO." to collectingRingHandler.

Passing "Log entry number 3 with level WARNING." to collectingRingHandler.

Passing "Log entry number 4 with level ERROR." to collectingRingHandler.

Passing "Log entry number 5 with level CRITICAL." to collectingRingHandler.

Passing "Log entry number 6 with level INFO." to collectingRingHandler.

Passing "Log entry number 7 with level ERROR." to collectingRingHandler.

Success Indexlist of log entries in stringrepresentation: Values and number of submitted values is correct. See detailed log for more information.

Result (Indexlist of log entries in stringrepresentation): [35, 109, 181, 256, 327] (<class 'list'>)

Expectation (Indexlist of log entries in stringrepresentation): result = [35, 109, 181, 256, 327] (<class 'list'>)

Result (Submitted value number 1): 35 (<class 'int'>)

Expectation (Submitted value number 1): result = 35 (<class 'int'>)

Submitted value number 1 is correct (Content 35 and Type is <class 'int'>).

Result (Submitted value number 2): 109 (<class 'int'>)

Expectation (Submitted value number 2): result = 109 (<class 'int'>)

Submitted value number 2 is correct (Content 109 and Type is <class 'int'>).

Result (Submitted value number 3): 181 (<class 'int'>)

Expectation (Submitted value number 3): result = 181 (<class 'int'>)

Submitted value number 3 is correct (Content 181 and Type is <class 'int'>).

Result (Submitted value number 4): 256 (<class 'int'>)

Expectation (Submitted value number 4): result = 256 (<class 'int'>)

Submitted value number 4 is correct (Content 256 and Type is <class 'int'>).

Result (Submitted value number 5): 327 (<class 'int'>)

Expectation (Submitted value number 5): result = 327 (<class 'int'>)

Submitted value number 5 is correct (Content 327 and Type is <class 'int'>).

B.1.5 Number of stored logs in the RingHandler

Description

The number of stored log-records shall be given on initialisation or reinitialisation.

Testresult

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

Info Running logger test sequence.

Configuring collecting logger

Passing "Log entry number 1 with level DEBUG." to collectingRingHandler.

Passing "Log entry number 2 with level INFO." to collectingRingHandler.

Passing "Log entry number 3 with level WARNING." to collectingRingHandler.

Passing "Log entry number 4 with level ERROR." to collectingRingHandler.

Passing "Log entry number 5 with level CRITICAL." to collectingRingHandler.

Passing "Log entry number 6 with level INFO." to collectingRingHandler.

Passing "Log entry number 7 with level ERROR." to collectingRingHandler.

Success Length of collectingRingHandler is correct (Content 5 and Type is <class 'int'>).

Result (Length of collectingRingHandler): 5 (<class 'int'>)

Expectation (Length of collectingRingHandler): result = 5 (<class 'int'>)

Success Length of collectingRingHandler after reinitialisation is correct (Content 3 and Type is <class 'int'>).

Result (Length of collectingRingHandler after reinitialisation): 3 (<class 'int'>)

Expectation (Length of collectingRingHandler after reinitialisation): result = 3 (<class 'int'>)

Success Log text is correct (Content 'Log entry number 5 with level CRITICAL.' and Type is <class 'str'>).

Result (Log text): 'Log entry number 5 with level CRITICAL.' (<class 'str'>)

Expectation (Log text): result = 'Log entry number 5 with level CRITICAL.' (<class 'str'>)

Success Log text is correct (Content 'Log entry number 6 with level INFO.' and Type is <class 'str'>).

Result (Log text): 'Log entry number 6 with level INFO.' (<class 'str'>)

Expectation (Log text): result = 'Log entry number 6 with level INFO.' (<class 'str'>)

Success Log text is correct (Content 'Log entry number 7 with level ERROR.' and Type is <class 'str'>).

Result (Log text): 'Log entry number 7 with level ERROR.' (<class 'str'>)

Expectation (Log text): result = 'Log entry number 7 with level ERROR.' (<class 'str'>)

C Test-Coverage

C.1 report

The line coverage for report was 89.9%

The branch coverage for report was 67.7%

C.1.1 report.__init__.py

The line coverage for report.__init__.py was 89.9%

The branch coverage for report.__init__.py was 67.7%

```

1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3 #
4 """
5 report (Report Module)
6 =====
7
8 **Author:**
9
10 * Dirk Alders <sudo-dirk@mount-mockery.de>
11
12 **Description:**
13
14     The Module is designed to help with python logging and to support some handlers for logging
15     to memory.
16
17 **Submodules:**
18
19 * :class:`report.collectingHandler`
20 * :class:`report.collectingRingHandler`
21 * :class:`report.collectingTestcaseHandler`
22 * :func:`report.consoleLoggingConfigure`
23 * :class:`report.testSession`
24
25 **Unittest:**
26
27     See also the :download:`unittest <../../report/_testresults_/unittest.pdf>` documentation
28 """
29 __DEPENDENCIES__ = []
30
31 import collections
32 import logging
33 from logging.config import dictConfig
34 import os
35 import sys
36
37 logger_name = 'REPORT'
38 logger = logging.getLogger(logger_name)
39
40 __DESCRIPTION__ = """The Module {\\tt %s} is designed to help with python logging and to support
41     some handlers for logging to memory.
42 For more Information read the sphinx documentation.""" % __name__.replace('-', '\\_')
43 """The Module Description"""
44 __INTERPRETER__ = (2, 3, )
45 """The Tested Interpreter-Versions"""
46
47 SHORT_FMT = "%(asctime)s: %(levelname)-8s- %(message)s"
48 """ A short formatter including the most important information """
49 LONG_FMT = """
50     ~~~~~~
51     File "%(pathname)s", line %(lineno)d, in %(funcName)s
52     %(asctime)s: %(levelname)-8s- %(message)s"""
53 """ A long formatter which results in links to the source code inside Eclipse"""
54 MAX_FMT = """

```


Unittest for report

```
53 %(name)s
54 %(levelname)s
55 %(levelno)s
56 %(pathname)s
57 %(filename)s
58 %(module)s
59 %(lineno)d
60 %(funcName)s
61 %(created)f
62 %(asctime)s
63 %(msecs)d
64 %(relativeCreated)d
65 %(thread)d
66 %(threadName)s
67 %(process)d
68 %(message)s"""
69 DEFAULT_FMT = LONG_FMT
70 """The default formatstring"""
71
72
73 class collectingHandler(logging.Handler):
74     MY_LOGS = []
75
76     def __init__(self):
77         logging.Handler.__init__(self)
78         self.setFormatter(logging.Formatter(MAX_FMT))
79         self.setLevel(logging.DEBUG)
80
81     def emit(self, record):
82         self.format(record)
83         self.MY_LOGS.append(record.__dict__)
84
85     def make_independent(self):
86         self.MY_LOGS = []
87
88     def get_logs(self):
89         rv = []
90         while len(self.MY_LOGS) > 0:
91             rv.append(self.MY_LOGS.pop(0))
92         return rv
93
94     def get_str(self, logs=None, fmt=SHORT_FMT):
95         logs = logs or self.MY_LOGS
96         return '\n'.join([fmt % log for log in logs])
97
98     def __len__(self):
99         return len(self.MY_LOGS)
100
101     def __str__(self):
102         return self.get_str(self.MY_LOGS)
103
104
105 class collectingRingHandler(collectingHandler):
106     MY_LOGS = collections.deque([], 10)
107
108     def __init__(self, max_logs=None):
109         collectingHandler.__init__(self)
110         if max_logs is not None and max_logs != self.MY_LOGS.maxlen:
111             self.MY_LOGS.__init__(list(self.MY_LOGS), max_logs)
112
113     def make_independent(self):
114         self.MY_LOGS = collections.deque([], self.MY_LOGS.maxlen)
115
```

Unittest for report

```
116     def get_logs(self):
117         return list(self.MY_LOGS)
118
119
120 TCEL_SINGLE = 0
121 """ Testcase level (smoke), this is just a rough test for the main functionality"""
122 TCEL_SMOKE = 10
123 """ Testcase level (smoke), this is just a rough test for the main functionality"""
124 TCEL_SHORT = 50
125 """ Testcase level (short), this is a short test for an extended functionality"""
126 TCEL_FULL = 90
127 """ Testcase level (full), this is a complete test for the full functionality"""
128 TCEL_NAMES = {
129     TCEL_SINGLE: 'Single Test',
130     TCEL_SMOKE: 'Smoke Test (Minumum subset)',
131     TCEL_SHORT: 'Short Test (Subset)',
132     TCEL_FULL: 'Full Test (all defined tests)'
133 }
134 """ Dictionary for resolving the test case levels (TCL) to a (human readable) name"""
135
136 TCEL_REVERSE_NAMED = {
137     'short': TCEL_SHORT,
138     'smoke': TCEL_SMOKE,
139     'single': TCEL_SINGLE,
140     'full': TCEL_FULL,
141 }
142 """ Dictionary for resolving named test case levels (TCL) to test case level number"""
143
144
145 class collectingTestcaseHandler(collectingHandler):
146     MY_LOGS = []
147
148     def emit(self, record):
149         self.format(record)
150         self.MY_LOGS.append(record.__dict__)
151         self.MY_LOGS[-1]['moduleLogger'] = collectingHandler().get_logs()
152
153
154 def consoleLoggingConfigure(loggers=[], level='DEBUG'):
155     my_loggers = {}
156     for l in loggers:
157         my_loggers[l] = dict(handlers=['console'], level=level, propagate=False)
158     #
159     logging_config = dict(
160         version=1,
161         formatters={
162             'short': {
163                 'format': SHORT_FMT,
164             },
165             'long': {
166                 'format': LONG_FMT,
167             },
168         },
169         handlers={
170             'console': {
171                 'level': level,
172                 'class': 'logging.StreamHandler',
173                 'formatter': 'short',
174                 'stream': 'ext://sys.stdout',
175             },
176         },
177         loggers=my_loggers,
178     )
```

Unittest for report

```
179 dictConfig(logging_config)
180
181
182 class testSession(dict):
183     KEY_NAME = 'name'
184     KEY_FAILED_TESTS = 'number_of_failed_tests'
185     KEY_POSSIBLY_FAILED_TESTS = 'number_of_possibly_failed_tests'
186     KEY_SUCCESS_TESTS = 'number_of_successful_tests'
187     KEY_ALL_TESTS = 'number_of_tests'
188     KEY_EXEC_LVL = 'testcase_execution_level'
189     KEY_EXEC_NAMES = 'testcase_names'
190     KEY_LVL_NAMES = 'level_names'
191     KEY_TESTCASELIST = 'testcases'
192     KEY_UID_LIST = 'uid_list_sorted'
193     #
194     DEFAULT_BASE_DATA = {
195         KEY_NAME: 'Default Testsession name',
196         KEY_FAILED_TESTS: 0,
197         KEY_POSSIBLY_FAILED_TESTS: 0,
198         KEY_FAILED_TESTS: 0,
199         KEY_SUCCESS_TESTS: 0,
200         KEY_ALL_TESTS: 0,
201         KEY_EXEC_LVL: TCEL_FULL,
202         KEY_EXEC_NAMES: TCEL_NAMES,
203     }
204
205     def __init__(self, module_names=[], **kwargs):
206         dict.__init__(self, time_consumption=0.)
207         self.__testcase__ = None
208         self.__set_base_data__(**kwargs)
209         self.__configure_logging__(module_names)
210
211     def __set_base_data__(self, **kwargs):
212         for key in set([key for key in self.DEFAULT_BASE_DATA.keys()] + [key for key in kwargs.
213             keys()]):
214             self[key] = kwargs.get(key, self.DEFAULT_BASE_DATA.get(key))
215         self[self.KEY_TESTCASELIST] = {}
216         self[self.KEY_UID_LIST] = []
217
218     def __configure_logging__(self, module_names):
219         #
220         # Configure logging for testSession
221         #
222         logging_config = dict(
223             version=1,
224             formatters={
225                 'short': {
226                     'format': SHORT_FMT,
227                 },
228                 'long': {
229                     'format': LONG_FMT,
230                 },
231             },
232             handlers={
233                 'console': {
234                     'level': 'DEBUG',
235                     'class': 'logging.NullHandler',
236                     'formatter': 'short',
237                 },
238                 'module_logs': {
239                     'level': 'DEBUG',
240                     'class': 'report.collectingHandler',
```

Unittest for report

```

240         'formatter': 'short',
241     },
242     'testcase_logs': {
243         'level': 'DEBUG',
244         'class': 'report.collectingTestcaseHandler',
245         'formatter': 'short',
246     },
247 },
248     loggers=self.__module_loggers__(module_names),
249 )
250 dictConfig(logging_config)
251
252 def __module_loggers__(self, module_names):
253     rv = {}
254     rv['__tLogger__'] = dict(handlers=['console', 'testcase_logs'], level='DEBUG', propagate=False)
255     for name in module_names + ['__mLogger__']:
256         rv[name] = dict(handlers=['console', 'module_logs'], level='DEBUG', propagate=False)
257     return rv
258
259 def testCase(self, name, testcase_execution_level, test_method, *args, **kwargs):
260     if testcase_execution_level <= self[self.KEY_EXEC_LVL]:
261         tLogger = logging.getLogger('__tLogger__')
262         tHandler = collectingTestcaseHandler()
263         if len(tHandler.MY_LOGS) > 0:
264             raise AttributeError("Testcaselgger shall be empty after closing testcase!")
265         tLogger._log(logging.DEBUG, name, None)
266         if len(tHandler.MY_LOGS) != 1:
267             raise AttributeError("Testcaselgger shall have only one entry for the main
268 testcase (temporary)!")
269         self.__testcase__ = tHandler.get_logs()[0]
270         test_method(logging.getLogger('__tLogger__'), *args, **kwargs)
271         self.__close_active_testcase__()
272
273 def __close_active_testcase__(self):
274     if self.__testcase__ is not None:
275         name = self.__testcase__.get('message')
276         #
277         # Add testcase
278         #
279         tch = collectingTestcaseHandler()
280         self.__testcase__[ 'testcaseLogger' ] = tch.get_logs()
281         if name in self[self.KEY_TESTCASELIST]:
282             raise AttributeError("Testcase named %s already exists" % name)
283         self[self.KEY_TESTCASELIST][name] = self.__testcase__
284         self[self.KEY_UID_LIST].append(name)
285         #
286         # Adapt testcase data
287         #
288         self[self.KEY_TESTCASELIST][name][ 'levelNo' ] = 0
289         self[self.KEY_TESTCASELIST][name][ 'time_consumption' ] = 0.
290         for teststep in self[self.KEY_TESTCASELIST][name][ 'testcaseLogger' ]:
291             # store maximum level to testcase
292             if teststep.get('levelNo') > self[self.KEY_TESTCASELIST][name][ 'levelNo' ]:
293                 self[self.KEY_TESTCASELIST][name][ 'levelNo' ] = teststep.get('levelNo')
294                 self[self.KEY_TESTCASELIST][name][ 'levelname' ] = teststep.get('levelname')
295             # store time_consumption for teststep
296             try:
297                 teststep[ 'time_consumption' ] = teststep[ 'created' ] - teststep[ 'moduleLogger'
298 ][-1][ 'created' ]
299             except IndexError:
300                 teststep[ 'time_consumption' ] = 0.

```

Unittest for report

```
299         # Increment testcase time_consumption
300     # Increment testcase counters
301     #
302     self[self.KEY_ALL_TESTS] += 1
303     if self[self.KEY_TESTCASELIST][name]['levelno'] <= logging.INFO:
304         self[self.KEY_SUCCESS_TESTS] += 1
305     elif self[self.KEY_TESTCASELIST][name]['levelno'] >= logging.ERROR:
306         self[self.KEY_FAILED_TESTS] += 1
307     else:
308         self[self.KEY_POSSIBLY_FAILED_TESTS] += 1
309     # Set testcase time and time_consumption
310     self[self.KEY_TESTCASELIST][name]['time_start'] = self.__testcase__['asctime']
311     self[self.KEY_TESTCASELIST][name]['time_finished'] = teststep['asctime']
312     self[self.KEY_TESTCASELIST][name]['time_consumption'] = teststep['created'] - self.
313     __testcase__['created']
314     # Set testcase time consumption
315     self['time_consumption'] += self[self.KEY_TESTCASELIST][name]['time_consumption']
316     self.__testcase__ = None
```