

Unittest for report

December 21, 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 2.7.18 (final) | 3 |
| 2.2 | Test-Statistic for testrun with python 3.8.5 (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 2.7.18 (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 3.8.5 (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 | 8f7c85c45e72c364342dcebaf79cd1d2 |

Dependencies

1.2 Unittest Information

Unittest Information

| | |
|---------------|---|
| Version | 2c03d3eba161a9fb0dbf0594fbda3965 |
| Testruns with | python 2.7.18 (final), python 3.8.5 (final) |

1.3 Test System Information

System Information

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

2 Statistic

2.1 Test-Statistic for testrun with python 2.7.18 (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 | 0.013s |

2.2 Test-Statistic for testrun with python 3.8.5 (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 |

2.3 Coverage Statistic

| Module- or Filename | Line-Coverage | Branch-Coverage |
|----------------------------|----------------------|------------------------|
| report | 81.0% | 51.2% |
| report.__init__.py | 81.0% | |

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 2.7.18 (final) |
| Caller: | /user_data/data/dirk/prj/unittest/report/unittest/src/tests/_init_.py (19) |
| Start-Time: | 2020-12-21 01:18:51,702 |
| Finished-Time: | 2020-12-21 01:18:51,705 |
| 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 3.8.5 (final) |
| Caller: | /user_data/data/dirk/prj/unittest/report/unittest/src/tests/_init_.py (19) |

Unittest for report

Start-Time: 2020-12-21 01:18:52,058
Finished-Time: 2020-12-21 01:18:52,060
Time-Consumption 0.002s

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 2.7.18 (final)
Caller: /user_data/data/dirk/prj/unittest/report/unittest/src/tests/...init...py (20)
Start-Time: 2020-12-21 01:18:51,705
Finished-Time: 2020-12-21 01:18:51,707
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 3.8.5 (final)
Caller: /user_data/data/dirk/prj/unittest/report/unittest/src/tests/...init...py (20)
Start-Time: 2020-12-21 01:18:52,060
Finished-Time: 2020-12-21 01:18:52,062
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.

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 2.7.18 (final) |
| Caller: | /user_data/data/dirk/prj/unittest/report/unittest/src/tests/_init_.py (21) |
| Start-Time: | 2020-12-21 01:18:51,707 |
| Finished-Time: | 2020-12-21 01:18:51,709 |
| 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 3.8.5 (final) |
| Caller: | /user_data/data/dirk/prj/unittest/report/unittest/src/tests/_init_.py (21) |
| Start-Time: | 2020-12-21 01:18:52,062 |
| Finished-Time: | 2020-12-21 01:18:52,064 |
| Time-Consumption | 0.002s |

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 2.7.18 (final) |
| Caller: | /user_data/data/dirk/prj/unittest/report/unittest/src/tests/_init_.py (22) |
| Start-Time: | 2020-12-21 01:18:51,710 |
| Finished-Time: | 2020-12-21 01:18:51,712 |
| 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 3.8.5 (final) |
| Caller: | /user_data/data/dirk/prj/unittest/report/unittest/src/tests/_init_.py (22) |
| Start-Time: | 2020-12-21 01:18:52,064 |
| Finished-Time: | 2020-12-21 01:18:52,066 |
| 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.

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 2.7.18 (final) |
| Caller: | /user_data/data/dirk/prj/unittest/report/unittest/src/tests/_init_.py (23) |
| Start-Time: | 2020-12-21 01:18:51,712 |

Unittest for report

Finished-Time: 2020-12-21 01:18:51,716
Time-Consumption 0.004s

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 3.8.5 (final) |
| Caller: | /user_data/data/dirk/prj/unittest/report/unittest/src/tests/_init_.py (23) |
| Start-Time: | 2020-12-21 01:18:52,066 |
| Finished-Time: | 2020-12-21 01:18:52,069 |
| Time-Consumption | 0.003s |

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 2.7.18 (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): [49, 134, 221, 309, 398, 486, 571
↪] (<type 'list'>)

Expectation (Indexlist of log entries in stringrepresentation): result = [49, 134, 221, 309,
↪ 398, 486, 571] (<type 'list'>)

Result (Submitted value number 1): 49 (<type 'int'>)

Expectation (Submitted value number 1): result = 49 (<type 'int'>)

Submitted value number 1 is correct (Content 49 and Type is <type 'int'>).

Result (Submitted value number 2): 134 (<type 'int'>)

Expectation (Submitted value number 2): result = 134 (<type 'int'>)

Submitted value number 2 is correct (Content 134 and Type is <type 'int'>).

Result (Submitted value number 3): 221 (<type 'int'>)

Expectation (Submitted value number 3): result = 221 (<type 'int'>)

Submitted value number 3 is correct (Content 221 and Type is <type 'int'>).

Result (Submitted value number 4): 309 (<type 'int'>)

Expectation (Submitted value number 4): result = 309 (<type 'int'>)

Submitted value number 4 is correct (Content 309 and Type is <type 'int'>).

Result (Submitted value number 5): 398 (<type 'int'>)

Expectation (Submitted value number 5): result = 398 (<type 'int'>)

Submitted value number 5 is correct (Content 398 and Type is <type 'int'>).

Result (Submitted value number 6): 486 (<type 'int'>)

Expectation (Submitted value number 6): result = 486 (<type 'int'>)

Submitted value number 6 is correct (Content 486 and Type is <type 'int'>).

Result (Submitted value number 7): 571 (<type 'int'>)

Expectation (Submitted value number 7): result = 571 (<type 'int'>)

Submitted value number 7 is correct (Content 571 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): [ 51, 139, 228, 316, 401 ] (<type 'list'>)
↳ 'list'>)
Expectation (Indexlist of log entries in stringrepresentation): result = [ 51, 139, 228, 316,
↳ 401 ] (<type 'list'>)
Result (Submitted value number 1): 51 (<type 'int'>)
Expectation (Submitted value number 1): result = 51 (<type 'int'>)
Submitted value number 1 is correct (Content 51 and Type is <type 'int'>).
Result (Submitted value number 2): 139 (<type 'int'>)
Expectation (Submitted value number 2): result = 139 (<type 'int'>)
Submitted value number 2 is correct (Content 139 and Type is <type 'int'>).
Result (Submitted value number 3): 228 (<type 'int'>)
Expectation (Submitted value number 3): result = 228 (<type 'int'>)
Submitted value number 3 is correct (Content 228 and Type is <type 'int'>).
Result (Submitted value number 4): 316 (<type 'int'>)
Expectation (Submitted value number 4): result = 316 (<type 'int'>)
Submitted value number 4 is correct (Content 316 and Type is <type 'int'>).
Result (Submitted value number 5): 401 (<type 'int'>)
Expectation (Submitted value number 5): result = 401 (<type 'int'>)
Submitted value number 5 is correct (Content 401 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 3.8.5 (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): [49, 134, 221, 309, 398, 486, 571
 ↪] (<class 'list'>)

Expectation (Indexlist of log entries in stringrepresentation): result = [49, 134, 221, 309,
 ↪ 398, 486, 571] (<class 'list'>)

Result (Submitted value number 1): 49 (<class 'int'>)

Expectation (Submitted value number 1): result = 49 (<class 'int'>)

Submitted value number 1 is correct (Content 49 and Type is <class 'int'>).

Result (Submitted value number 2): 134 (<class 'int'>)

Expectation (Submitted value number 2): result = 134 (<class 'int'>)

Submitted value number 2 is correct (Content 134 and Type is <class 'int'>).

Result (Submitted value number 3): 221 (<class 'int'>)

Expectation (Submitted value number 3): result = 221 (<class 'int'>)

Submitted value number 3 is correct (Content 221 and Type is <class 'int'>).

Result (Submitted value number 4): 309 (<class 'int'>)

Expectation (Submitted value number 4): result = 309 (<class 'int'>)

Submitted value number 4 is correct (Content 309 and Type is <class 'int'>).

Result (Submitted value number 5): 398 (<class 'int'>)

Expectation (Submitted value number 5): result = 398 (<class 'int'>)

Submitted value number 5 is correct (Content 398 and Type is <class 'int'>).

Result (Submitted value number 6): 486 (<class 'int'>)

Expectation (Submitted value number 6): result = 486 (<class 'int'>)

Submitted value number 6 is correct (Content 486 and Type is <class 'int'>).

Result (Submitted value number 7): 571 (<class 'int'>)

Expectation (Submitted value number 7): result = 571 (<class 'int'>)

Submitted value number 7 is correct (Content 571 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): [51, 139, 228, 316, 401] (<class 'list'>)

Expectation (Indexlist of log entries in stringrepresentation): result = [51, 139, 228, 316, 401] (<class 'list'>)

Result (Submitted value number 1): 51 (<class 'int'>)

Expectation (Submitted value number 1): result = 51 (<class 'int'>)

Submitted value number 1 is correct (Content 51 and Type is <class 'int'>).

Result (Submitted value number 2): 139 (<class 'int'>)

Expectation (Submitted value number 2): result = 139 (<class 'int'>)

Submitted value number 2 is correct (Content 139 and Type is <class 'int'>).

Result (Submitted value number 3): 228 (<class 'int'>)

Expectation (Submitted value number 3): result = 228 (<class 'int'>)

Submitted value number 3 is correct (Content 228 and Type is <class 'int'>).

Result (Submitted value number 4): 316 (<class 'int'>)

Expectation (Submitted value number 4): result = 316 (<class 'int'>)

Submitted value number 4 is correct (Content 316 and Type is <class 'int'>).

Result (Submitted value number 5): 401 (<class 'int'>)

Expectation (Submitted value number 5): result = 401 (<class 'int'>)

Submitted value number 5 is correct (Content 401 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 81.0%

The branch coverage for report was 51.2%

C.1.1 report.__init__.py

The line coverage for report.__init__.py was 81.0%

The branch coverage for report.__init__.py was 51.2%

```

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 json
33 import logging
34 from logging.config import dictConfig
35 import os
36 import sys
37
38 try:
39     from config import APP_NAME as ROOT_LOGGER_NAME
40 except ImportError:
41     ROOT_LOGGER_NAME = 'root'
42 logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)
43
44 __DESCRIPTION__ = """The Module {\\tt %s} is designed to help with python logging and to support
45     some handlers for logging to memory.
46 For more Information read the sphinx documentation.""" % __name__.replace('-', '\\-')
47 """The Module Description"""
48 __INTERPRETER__ = (2, 3, )
49 """The Tested Interpreter-Versions"""
50
51 SHORT_FMT = "%(asctime)s: %(name)s - %(levelname)s - %(message)s"
52 """ A short formatter including the most important information"""
53 LONG_FMT = """~~~~~%(levelname)-10s
54 ~~~~~~
55 ~~~~~~
56 ~~~~~~
57 ~~~~~~
58 ~~~~~~
59 ~~~~~~
60 ~~~~~~
61 ~~~~~~
62 ~~~~~~
63 ~~~~~~
64 ~~~~~~
65 ~~~~~~
66 ~~~~~~
67 ~~~~~~
68 ~~~~~~
69 ~~~~~~
70 ~~~~~~
71 ~~~~~~
72 ~~~~~~
73 ~~~~~~
74 ~~~~~~
75 ~~~~~~
76 ~~~~~~
77 ~~~~~~
78 ~~~~~~
79 ~~~~~~
80 ~~~~~~
81 ~~~~~~
82 ~~~~~~
83 ~~~~~~
84 ~~~~~~
85 ~~~~~~
86 ~~~~~~
87 ~~~~~~
88 ~~~~~~
89 ~~~~~~
90 ~~~~~~
91 ~~~~~~
92 ~~~~~~
93 ~~~~~~
94 ~~~~~~
95 ~~~~~~
96 ~~~~~~
97 ~~~~~~
98 ~~~~~~
99 ~~~~~~
100 ~~~~~~
101 ~~~~~~
102 ~~~~~~
103 ~~~~~~
104 ~~~~~~
105 ~~~~~~
106 ~~~~~~
107 ~~~~~~
108 ~~~~~~
109 ~~~~~~
110 ~~~~~~
111 ~~~~~~
112 ~~~~~~
113 ~~~~~~
114 ~~~~~~
115 ~~~~~~
116 ~~~~~~
117 ~~~~~~
118 ~~~~~~
119 ~~~~~~
120 ~~~~~~
121 ~~~~~~
122 ~~~~~~
123 ~~~~~~
124 ~~~~~~
125 ~~~~~~
126 ~~~~~~
127 ~~~~~~
128 ~~~~~~
129 ~~~~~~
130 ~~~~~~
131 ~~~~~~
132 ~~~~~~
133 ~~~~~~
134 ~~~~~~
135 ~~~~~~
136 ~~~~~~
137 ~~~~~~
138 ~~~~~~
139 ~~~~~~
140 ~~~~~~
141 ~~~~~~
142 ~~~~~~
143 ~~~~~~
144 ~~~~~~
145 ~~~~~~
146 ~~~~~~
147 ~~~~~~
148 ~~~~~~
149 ~~~~~~
150 ~~~~~~
151 ~~~~~~
152 ~~~~~~
153 ~~~~~~
154 ~~~~~~
155 ~~~~~~
156 ~~~~~~
157 ~~~~~~
158 ~~~~~~
159 ~~~~~~
160 ~~~~~~
161 ~~~~~~
162 ~~~~~~
163 ~~~~~~
164 ~~~~~~
165 ~~~~~~
166 ~~~~~~
167 ~~~~~~
168 ~~~~~~
169 ~~~~~~
170 ~~~~~~
171 ~~~~~~
172 ~~~~~~
173 ~~~~~~
174 ~~~~~~
175 ~~~~~~
176 ~~~~~~
177 ~~~~~~
178 ~~~~~~
179 ~~~~~~
180 ~~~~~~
181 ~~~~~~
182 ~~~~~~
183 ~~~~~~
184 ~~~~~~
185 ~~~~~~
186 ~~~~~~
187 ~~~~~~
188 ~~~~~~
189 ~~~~~~
190 ~~~~~~
191 ~~~~~~
192 ~~~~~~
193 ~~~~~~
194 ~~~~~~
195 ~~~~~~
196 ~~~~~~
197 ~~~~~~
198 ~~~~~~
199 ~~~~~~
200 ~~~~~~
201 ~~~~~~
202 ~~~~~~
203 ~~~~~~
204 ~~~~~~
205 ~~~~~~
206 ~~~~~~
207 ~~~~~~
208 ~~~~~~
209 ~~~~~~
210 ~~~~~~
211 ~~~~~~
212 ~~~~~~
213 ~~~~~~
214 ~~~~~~
215 ~~~~~~
216 ~~~~~~
217 ~~~~~~
218 ~~~~~~
219 ~~~~~~
220 ~~~~~~
221 ~~~~~~
222 ~~~~~~
223 ~~~~~~
224 ~~~~~~
225 ~~~~~~
226 ~~~~~~
227 ~~~~~~
228 ~~~~~~
229 ~~~~~~
230 ~~~~~~
231 ~~~~~~
232 ~~~~~~
233 ~~~~~~
234 ~~~~~~
235 ~~~~~~
236 ~~~~~~
237 ~~~~~~
238 ~~~~~~
239 ~~~~~~
240 ~~~~~~
241 ~~~~~~
242 ~~~~~~
243 ~~~~~~
244 ~~~~~~
245 ~~~~~~
246 ~~~~~~
247 ~~~~~~
248 ~~~~~~
249 ~~~~~~
250 ~~~~~~
251 ~~~~~~
252 ~~~~~~
253 ~~~~~~
254 ~~~~~~
255 ~~~~~~
256 ~~~~~~
257 ~~~~~~
258 ~~~~~~
259 ~~~~~~
260 ~~~~~~
261 ~~~~~~
262 ~~~~~~
263 ~~~~~~
264 ~~~~~~
265 ~~~~~~
266 ~~~~~~
267 ~~~~~~
268 ~~~~~~
269 ~~~~~~
270 ~~~~~~
271 ~~~~~~
272 ~~~~~~
273 ~~~~~~
274 ~~~~~~
275 ~~~~~~
276 ~~~~~~
277 ~~~~~~
278 ~~~~~~
279 ~~~~~~
280 ~~~~~~
281 ~~~~~~
282 ~~~~~~
283 ~~~~~~
284 ~~~~~~
285 ~~~~~~
286 ~~~~~~
287 ~~~~~~
288 ~~~~~~
289 ~~~~~~
290 ~~~~~~
291 ~~~~~~
292 ~~~~~~
293 ~~~~~~
294 ~~~~~~
295 ~~~~~~
296 ~~~~~~
297 ~~~~~~
298 ~~~~~~
299 ~~~~~~
300 ~~~~~~
301 ~~~~~~
302 ~~~~~~
303 ~~~~~~
304 ~~~~~~
305 ~~~~~~
306 ~~~~~~
307 ~~~~~~
308 ~~~~~~
309 ~~~~~~
310 ~~~~~~
311 ~~~~~~
312 ~~~~~~
313 ~~~~~~
314 ~~~~~~
315 ~~~~~~
316 ~~~~~~
317 ~~~~~~
318 ~~~~~~
319 ~~~~~~
320 ~~~~~~
321 ~~~~~~
322 ~~~~~~
323 ~~~~~~
324 ~~~~~~
325 ~~~~~~
326 ~~~~~~
327 ~~~~~~
328 ~~~~~~
329 ~~~~~~
330 ~~~~~~
331 ~~~~~~
332 ~~~~~~
333 ~~~~~~
334 ~~~~~~
335 ~~~~~~
336 ~~~~~~
337 ~~~~~~
338 ~~~~~~
339 ~~~~~~
340 ~~~~~~
341 ~~~~~~
342 ~~~~~~
343 ~~~~~~
344 ~~~~~~
345 ~~~~~~
346 ~~~~~~
347 ~~~~~~
348 ~~~~~~
349 ~~~~~~
350 ~~~~~~
351 ~~~~~~
352 ~~~~~~
353 ~~~~~~
354 ~~~~~~
355 ~~~~~~
356 ~~~~~~
357 ~~~~~~
358 ~~~~~~
359 ~~~~~~
360 ~~~~~~
361 ~~~~~~
362 ~~~~~~
363 ~~~~~~
364 ~~~~~~
365 ~~~~~~
366 ~~~~~~
367 ~~~~~~
368 ~~~~~~
369 ~~~~~~
370 ~~~~~~
371 ~~~~~~
372 ~~~~~~
373 ~~~~~~
374 ~~~~~~
375 ~~~~~~
376 ~~~~~~
377 ~~~~~~
378 ~~~~~~
379 ~~~~~~
380 ~~~~~~
381 ~~~~~~
382 ~~~~~~
383 ~~~~~~
384 ~~~~~~
385 ~~~~~~
386 ~~~~~~
387 ~~~~~~
388 ~~~~~~
389 ~~~~~~
390 ~~~~~~
391 ~~~~~~
392 ~~~~~~
393 ~~~~~~
394 ~~~~~~
395 ~~~~~~
396 ~~~~~~
397 ~~~~~~
398 ~~~~~~
399 ~~~~~~
400 ~~~~~~
401 ~~~~~~
402 ~~~~~~
403 ~~~~~~
404 ~~~~~~
405 ~~~~~~
406 ~~~~~~
407 ~~~~~~
408 ~~~~~~
409 ~~~~~~
410 ~~~~~~
411 ~~~~~~
412 ~~~~~~
413 ~~~~~~
414 ~~~~~~
415 ~~~~~~
416 ~~~~~~
417 ~~~~~~
418 ~~~~~~
419 ~~~~~~
420 ~~~~~~
421 ~~~~~~
422 ~~~~~~
423 ~~~~~~
424 ~~~~~~
425 ~~~~~~
426 ~~~~~~
427 ~~~~~~
428 ~~~~~~
429 ~~~~~~
430 ~~~~~~
431 ~~~~~~
432 ~~~~~~
433 ~~~~~~
434 ~~~~~~
435 ~~~~~~
436 ~~~~~~
437 ~~~~~~
438 ~~~~~~
439 ~~~~~~
440 ~~~~~~
441 ~~~~~~
442 ~~~~~~
443 ~~~~~~
444 ~~~~~~
445 ~~~~~~
446 ~~~~~~
447 ~~~~~~
448 ~~~~~~
449 ~~~~~~
450 ~~~~~~
451 ~~~~~~
452 ~~~~~~
453 ~~~~~~
454 ~~~~~~
455 ~~~~~~
456 ~~~~~~
457 ~~~~~~
458 ~~~~~~
459 ~~~~~~
460 ~~~~~~
461 ~~~~~~
462 ~~~~~~
463 ~~~~~~
464 ~~~~~~
465 ~~~~~~
466 ~~~~~~
467 ~~~~~~
468 ~~~~~~
469 ~~~~~~
470 ~~~~~~
471 ~~~~~~
472 ~~~~~~
473 ~~~~~~
474 ~~~~~~
475 ~~~~~~
476 ~~~~~~
477 ~~~~~~
478 ~~~~~~
479 ~~~~~~
480 ~~~~~~
481 ~~~~~~
482 ~~~~~~
483 ~~~~~~
484 ~~~~~~
485 ~~~~~~
486 ~~~~~~
487 ~~~~~~
488 ~~~~~~
489 ~~~~~~
490 ~~~~~~
491 ~~~~~~
492 ~~~~~~
493 ~~~~~~
494 ~~~~~~
495 ~~~~~~
496 ~~~~~~
497 ~~~~~~
498 ~~~~~~
499 ~~~~~~
500 ~~~~~~
501 ~~~~~~
502 ~~~~~~
503 ~~~~~~
504 ~~~~~~
505 ~~~~~~
506 ~~~~~~
507 ~~~~~~
508 ~~~~~~
509 ~~~~~~
510 ~~~~~~
511 ~~~~~~
512 ~~~~~~
513 ~~~~~~
514 ~~~~~~
515 ~~~~~~
516 ~~~~~~
517 ~~~~~~
518 ~~~~~~
519 ~~~~~~
520 ~~~~~~
521 ~~~~~~
522 ~~~~~~
523 ~~~~~~
524 ~~~~~~
525 ~~~~~~
526 ~~~~~~
527 ~~~~~~
528 ~~~~~~
529 ~~~~~~
530 ~~~~~~
531 ~~~~~~
532 ~~~~~~
533 ~~~~~~
534 ~~~~~~
535 ~~~~~~
536 ~~~~~~
537 ~~~~~~
538 ~~~~~~
539 ~~~~~~
540 ~~~~~~
541 ~~~~~~
542 ~~~~~~
543 ~~~~~~
544 ~~~~~~
545 ~~~~~~
546 ~~~~~~
547 ~~~~~~
548 ~~~~~~
549 ~~~~~~
550 ~~~~~~
551 ~~~~~~
552 ~~~~~~
553 ~~~~~~
554 ~~~~~~
555 ~~~~~~
556 ~~~~~~
557 ~~~~~~
558 ~~~~~~
559 ~~~~~~
560 ~~~~~~
561 ~~~~~~
562 ~~~~~~
563 ~~~~~~
564 ~~~~~~
565 ~~~~~~
566 ~~~~~~
567 ~~~~~~
568 ~~~~~~
569 ~~~~~~
570 ~~~~~~
571 ~~~~~~
572 ~~~~~~
573 ~~~~~~
574 ~~~~~~
575 ~~~~~~
576 ~~~~~~
577 ~~~~~~
578 ~~~~~~
579 ~~~~~~
580 ~~~~~~
581 ~~~~~~
582 ~~~~~~
583 ~~~~~~
584 ~~~~~~
585 ~~~~~~
586 ~~~~~~
587 ~~~~~~
588 ~~~~~~
589 ~~~~~~
590 ~~~~~~
591 ~~~~~~
592 ~~~~~~
593 ~~~~~~
594 ~~~~~~
595 ~~~~~~
596 ~~~~~~
597 ~~~~~~
598 ~~~~~~
599 ~~~~~~
600 ~~~~~~
601 ~~~~~~
602 ~~~~~~
603 ~~~~~~
604 ~~~~~~
605 ~~~~~~
606 ~~~~~~
607 ~~~~~~
608 ~~~~~~
609 ~~~~~~
610 ~~~~~~
611 ~~~~~~
612 ~~~~~~
613 ~~~~~~
614 ~~~~~~
615 ~~~~~~
616 ~~~~~~
617 ~~~~~~
618 ~~~~~~
619 ~~~~~~
620 ~~~~~~
621 ~~~~~~
622 ~~~~~~
623 ~~~~~~
624 ~~~~~~
625 ~~~~~~
626 ~~~~~~
627 ~~~~~~
628 ~~~~~~
629 ~~~~~~
630 ~~~~~~
631 ~~~~~~
632 ~~~~~~
633 ~~~~~~
634 ~~~~~~
635 ~~~~~~
636 ~~~~~~
637 ~~~~~~
638 ~~~~~~
639 ~~~~~~
640 ~~~~~~
641 ~~~~~~
642 ~~~~~~
643 ~~~~~~
644 ~~~~~~
645 ~~~~~~
646 ~~~~~~
647 ~~~~~~
648 ~~~~~~
649 ~~~~~~
650 ~~~~~~
651 ~~~~~~
652 ~~~~~~
653 ~~~~~~
654 ~~~~~~
655 ~~~~~~
656 ~~~~~~
657 ~~~~~~
658 ~~~~~~
659 ~~~~~~
660 ~~~~~~
661 ~~~~~~
662 ~~~~~~
663 ~~~~~~
664 ~~~~~~
665 ~~~~~~
666 ~~~~~~
667 ~~~~~~
668 ~~~~~~
669 ~~~~~~
670 ~~~~~~
671 ~~~~~~
672 ~~~~~~
673 ~~~~~~
674 ~~~~~~
675 ~~~~~~
676 ~~~~~~
677 ~~~~~~
678 ~~~~~~
679 ~~~~~~
680 ~~~~~~
681 ~~~~~~
682 ~~~~~~
683 ~~~~~~
684 ~~~~~~
685 ~~~~~~
686 ~~~~~~
687 ~~~~~~
688 ~~~~~~
689 ~~~~~~
690 ~~~~~~
691 ~~~~~~
692 ~~~~~~
693 ~~~~~~
694 ~~~~~~
695 ~~~~~~
696 ~~~~~~
697 ~~~~~~
698 ~~~~~~
699 ~~~~~~
700 ~~~~~~
701 ~~~~~~
702 ~~~~~~
703 ~~~~~~
704 ~~~~~~
705 ~~~~~~
706 ~~~~~~
707 ~~~~~~
708 ~~~~~~
709 ~~~~~~
710 ~~~~~~
711 ~~~~~~
712 ~~~~~~
713 ~~~~~~
714 ~~~~~~
715 ~~~~~~
716 ~~~~~~
717 ~~~~~~
718 ~~~~~~
719 ~~~~~~
720 ~~~~~~
721 ~~~~~~
722 ~~~~~~
723 ~~~~~~
724 ~~~~~~
725 ~~~~~~
726 ~~~~~~
727 ~~~~~~
728 ~~~~~~
729 ~~~~~~
730 ~~~~~~
731 ~~~~~~
732 ~~~~~~
733 ~~~~~~
734 ~~~~~~
735 ~~~~~~
736 ~~~~~~
737 ~~~~~~
738 ~~~~~~
739 ~~~~~~
740 ~~~~~~
741 ~~~~~~
742 ~~~~~~
743 ~~~~~~
744 ~~~~~~
745 ~~~~~~
746 ~~~~~~
747 ~~~~~~
748 ~~~~~~
749 ~~~~~~
750 ~~~~~~
751 ~~~~~~
752 ~~~~~~
753 ~~~~~~
754 ~~~~~~
755 ~~~~~~
756 ~~~~~~
757 ~~~~~~
758 ~~~~~~
759 ~~~~~~
760 ~~~~~~
761 ~~~~~~
762 ~~~~~~
763 ~~~~~~
764 ~~~~~~
765 ~~~~~~
766 ~~~~~~
767 ~~~~~~
768 ~~~~~~
769 ~~~~~~
770 ~~~~~~
771 ~~~~~~
772 ~~~~~~
773 ~~~~~~
774 ~~~~~~
775 ~~~~~~
776 ~~~~~~
777 ~~~~~~
778 ~~~~~~
779 ~~~~~~
780 ~~~~~~
781 ~~~~~~
782 ~~~~~~
783 ~~~~~~
784 ~~~~~~
785 ~~~~~~
786 ~~~~~~
787 ~~~~~~
788 ~~~~~~
789 ~~~~~~
790 ~~~~~~
791 ~~~~~~
792 ~~~~~~
793 ~~~~~~
794 ~~~~~~
795 ~~~~~~
796 ~~~~~~
797 ~~~~~~
798 ~~~~~~
799 ~~~~~~
800 ~~~~~~
801 ~~~~~~
802 ~~~~~~
803 ~~~~~~
804 ~~~~~~
805 ~~~~~~
806 ~~~~~~
807 ~~~~~~
808 ~~~~~~
809 ~~~~~~
810 ~~~~~~
811 ~~~~~~
812 ~~~~~~
813 ~~~~~~
814 ~~~~~~
815 ~~~~~~
816 ~~~~~~
817 ~~~~~~
818 ~~~~~~
819 ~~~~~~
820 ~~~~~~
821 ~~~~~~
822 ~~~~~~
823 ~~~~~~
824 ~~~~~~
825 ~~~~~~
826 ~~~~~~
827 ~~~~~~
828 ~~~~~~
829 ~~~~~~
830 ~~~~~~
831 ~~~~~~
832 ~~~~~~
833 ~~~~~~
834 ~~~~~~
835 ~~~~~~
836 ~~~~~~
837 ~~~~~~
838 ~~~~~~
839 ~~~~~~
840 ~~~~~~
841 ~~~~~~
842 ~~~~~~
843 ~~~~~~
844 ~~~~~~
845 ~~~~~~
846 ~~~~~~
847 ~~~~~~
848 ~~~~~~
849 ~~~~~~
850 ~~~~~~
851 ~~~~~~
852 ~~~~~~
853 ~~~~~~
854 ~~~~~~
855 ~~~~~~
856 ~~~~~~
857 ~~~~~~
858 ~~~~~~
859 ~~~~~~
860 ~~~~~~
861 ~~~~~~
862 ~~~~~~
863 ~~~~~~
864 ~~~~~~
865 ~~~~~~
866 ~~~~~~
867 ~~~~~~
868 ~~~~~~
869 ~~~~~~
870 ~~~~~~
871 ~~~~~~
872 ~~~~~~
873 ~~~~~~
874 ~~~~~~
875 ~~~~~~
876 ~~~~~~
877 ~~~~~~
878 ~~~~~~
879 ~~~~~~
880 ~~~~~~
881 ~~~~~~
882 ~~~~~~
883 ~~~~~~
884 ~~~~~~
885 ~~~~~~
886 ~~~~~~
887 ~~~~~~
888 ~~~~~~
889 ~~~~~~
890 ~~~~~~
891 ~~~~~~
892 ~~~~~~
893 ~~~~~~
894 ~~~~~~
895 ~~~~~~
896 ~~~~~~
897 ~~~~~~
898 ~~~~~~
899 ~~~~~~
900 ~~~~~~
901 ~~~~~~
902 ~~~~~~
903 ~~~~~~
904 ~~~~~~
905 ~~~~~~
906 ~~~~~~
907 ~~~~~~
908 ~~~~~~
909 ~~~~~~
910 ~~~~~~
911 ~~~~~~
912 ~~~~~~
913 ~~~~~~
914 ~~~~~~
915 ~~~~~~
916 ~~~~~~
917 ~~~~~~
918 ~~~~~~
919 ~~~~~~
920 ~~~~~~
921 ~~~~~~
922 ~~~~~~
923 ~~~~~~
924 ~~~~~~
925 ~~~~~~
926 ~~~~~~
927 ~~~~~~
928 ~~~~~~
929 ~~~~~~
930 ~~~~~~
931 ~~~~~~
932 ~~~~~~
933 ~~~~~~
934 ~~~~~~
935 ~~~~~~
936 ~~~~~~
937 ~~~~~~
938 ~~~~~~
939 ~~~~~~
940 ~~~~~~
941 ~~~~~~
942 ~~~~~~
943 ~~~~~~
944 ~~~~~~
945 ~~~~~~
946 ~~~~~~
947 ~~~~~~
948 ~~~~~~
949 ~~~~~~
950 ~~~~~~
951 ~~~~~~
952 ~~~~~~
953 ~~~~~~
954 ~~~~~~
955 ~~~~~~
956 ~~~~~~
957 ~~~~~~
958 ~~~~~~
959 ~~~~~~
960 ~~~~~~
961 ~~~~~~
962 ~~~~~~
963 ~~~~~~
964 ~~~~~~
965 ~~~~~~
966 ~~~~~~
967 ~~~~~~
968 ~~~~~~
969 ~~~~~~
970 ~~~~~~
971 ~~~~~~
972 ~~~~~~
973 ~~~~~~
974 ~~~~~~
975 ~~~~~~
976 ~~~~~~
977 ~~~~~~
978 ~~~~~~
979 ~~~~~~
980 ~~~~~~
981 ~~~~~~
982 ~~~~~~
983 ~~~~~~
984 ~~~~~~
985 ~~~~~~
986 ~~~~~~
987 ~~~~~~
988 ~~~~~~
989 ~~~~~~
990 ~~~~~~
991 ~~~~~~
992 ~~~~~~
993 ~~~~~~
994 ~~~~~~
995 ~~~~~~
996 ~~~~~~
997 ~~~~~~
998 ~~~~~~
999 ~~~~~~
1000 ~~~~~~

```


Unittest for report

```
53 File "%(pathname)s", line %(lineno)d, in %(funcName)s
54 %(asctime)s: %(name)s- %(message)s
55 ~~~~~
56 """ A long formatter which results in links to the source code inside Eclipse"""
57 MAX_FMT = """
58 %(name)s
59 %(levelno)s
60 %(levelname)s
61 %(pathname)s
62 %(filename)s
63 %(module)s
64 %(lineno)d
65 %(funcName)s
66 %(created)f
67 %(asctime)s
68 %(msecs)d
69 %(relativeCreated)d
70 %(thread)d
71 %(threadName)s
72 %(process)d
73 %(message)s"""
74 DEFAULT_FMT = LONG_FMT
75 """ The default formatstring"""
76
77
78 class collectingHandler(logging.Handler):
79     MY_LOGS = []
80
81     def __init__(self):
82         logging.Handler.__init__(self)
83         self.setFormatter(logging.Formatter(MAX_FMT))
84         self.setLevel(logging.DEBUG)
85
86     def emit(self, record):
87         self.format(record)
88         self.MY_LOGS.append(record.__dict__)
89
90     def make_independent(self):
91         self.MY_LOGS = []
92
93     def get_logs(self):
94         rv = []
95         while len(self.MY_LOGS) > 0:
96             rv.append(self.MY_LOGS.pop(0))
97         return rv
98
99     def get_str(self, logs=None, fmt=SHORT_FMT):
100         logs = logs or self.MY_LOGS
101         return '\n'.join([fmt % log for log in logs])
102
103     def __len__(self):
104         return len(self.MY_LOGS)
105
106     def __str__(self):
107         return self.get_str(self.MY_LOGS)
108
109
110 class collectingRingHandler(collectingHandler):
111     MY_LOGS = collections.deque([], 10)
112
```

Unittest for report

```
113     def __init__(self, max_logs=None):
114         collectingHandler.__init__(self)
115         if max_logs is not None and max_logs != self.MY_LOGS.maxlen:
116             self.MY_LOGS.__init__(list(self.MY_LOGS), max_logs)
117
118     def make_independent(self):
119         self.MY_LOGS = collections.deque([], self.MY_LOGS.maxlen)
120
121     def get_logs(self):
122         return list(self.MY_LOGS)
123
124
125 TCEL_SINGLE = 0
126 """ Testcase level (smoke), this is just a rough test for the main functionality"""
127 TCEL_SMOKE = 10
128 """ Testcase level (smoke), this is just a rough test for the main functionality"""
129 TCEL_SHORT = 50
130 """ Testcase level (short), this is a short test for an extended functionality"""
131 TCEL_FULL = 90
132 """ Testcase level (full), this is a complete test for the full functionality"""
133 TCEL_NAMES = {
134     TCEL_SINGLE: 'Single Test',
135     TCEL_SMOKE: 'Smoke Test (Minumum subset)',
136     TCEL_SHORT: 'Short Test (Subset)',
137     TCEL_FULL: 'Full Test (all defined tests)'
138 }
139 """ Dictionary for resolving the test case levels (TCL) to a (human readable) name"""
140
141 TCEL_REVERSE_NAMED = {
142     'short': TCEL_SHORT,
143     'smoke': TCEL_SMOKE,
144     'single': TCEL_SINGLE,
145     'full': TCEL_FULL,
146 }
147 """ Dictionary for resolving named test case levels (TCL) to test case level number"""
148
149
150 class collectingTestcaseHandler(collectingHandler):
151     MY_LOGS = []
152
153     def emit(self, record):
154         self.format(record)
155         self.MY_LOGS.append(record.__dict__)
156         self.MY_LOGS[-1]['moduleLogger'] = collectingHandler().get_logs()
157
158
159 class JsonFormatter(logging.Formatter):
160     def format(self, record):
161         obj = {}
162         for key in ["name", "levelno", "levelname", "pathname", "filename", "module", "lineno", "
163 funcName", "created", "msecs", "relativeCreated", "thread", "threadName", "process", "
164 processName", "msg", "args", "exc_info", "exc_text"]:
165             obj[key] = getattr(record, key)
166             obj["msg"] = obj["msg"] % obj["args"]
167             return json.dumps(obj)
168
169
170 def appLoggingConfigure(basepath, target, log_name_lvl=[], fmt=SHORT_FMT, ring_logs=None, host=
171 None, port=None):
172     target_handlers = ['main', ]
173     # define handler
174     #
```

Unittest for report

```
172 if target == 'stdout':
173     handler = dict(main={
174         'level': 'DEBUG',
175         'formatter': 'format',
176         'class': 'logging.StreamHandler',
177         'stream': 'ext://sys.stdout',
178     })
179 elif target == 'logfile':
180     handler = dict(main={
181         'level': 'DEBUG',
182         'formatter': 'json',
183         'class': 'logging.handlers.RotatingFileHandler',
184         'filename': os.path.join(basepath, 'messages.log'),
185         'mode': 'a',
186         'maxBytes': 10485760,
187         'backupCount': 7
188     })
189 else:
190     handler = dict(my_handler={
191         'level': 'DEBUG',
192         'formatter': 'my_format',
193         'class': 'logging.NullHandler',
194     })
195 if host is not None and port is not None:
196     target_handlers.append('socket')
197     handler['socket']={
198         'level': 'DEBUG',
199         'class': 'logging.handlers.SocketHandler',
200         'host': host,
201         'port': port
202     }
203 if ring_logs is not None:
204     target_handlers.append('ring')
205     handler['ring'] = {
206         'class': 'report.collectingRingHandler',
207         'max_logs': ring_logs,
208     }
209 # define loggers
210 #
211 loggers = {}
212 for name, lvl in log_name_lvl:
213     loggers[name] = {
214         'handlers': target_handlers,
215         'level': lvl,
216         'propagate': False
217     }
218 # configure logging
219 #
220 dictConfig(dict(
221     version=1,
222     formatters={
223         'json': {
224             '()': JsonFormatter
225         },
226         'long': {
227             'format': LONGFMT
228         },
229         'format': {
230             'format': fmt,
231         },
232     },
233     handlers=handler,
```

Unittest for report

```
234         loggers=loggers ,
235     ))
236
237
238 def stdoutLoggingConfigure(log_name_lvl=[], fmt=SHORT_FMT):
239     appLoggingConfigure(None, 'stdout', log_name_lvl=log_name_lvl, fmt=fmt)
240
241
242 class testSession(dict):
243     KEY_NAME = 'name'
244     KEY_FAILED_TESTS = 'number_of_failed_tests'
245     KEY_POSSIBLY_FAILED_TESTS = 'number_of_possibly_failed_tests'
246     KEY_SUCCESS_TESTS = 'number_of_successful_tests'
247     KEY_ALL_TESTS = 'number_of_tests'
248     KEY_EXEC_LVL = 'testcase_execution_level'
249     KEY_EXEC_NAMES = 'testcase_names'
250     KEY_LVL_NAMES = 'level_names'
251     KEY_TESTCASELIST = 'testcases'
252     KEY_UID_LIST = 'uid_list_sorted'
253     #
254     DEFAULT_BASE_DATA = {
255         KEY_NAME: 'Default Testsession name',
256         KEY_FAILED_TESTS: 0,
257         KEY_POSSIBLY_FAILED_TESTS: 0,
258         KEY_FAILED_TESTS: 0,
259         KEY_SUCCESS_TESTS: 0,
260         KEY_ALL_TESTS: 0,
261         KEY_EXEC_LVL: TCEL_FULL,
262         KEY_EXEC_NAMES: TCEL_NAMES,
263     }
264
265     def __init__(self, module_names=[], **kwargs):
266         dict.__init__(self, time_consumption=0.)
267         self.__testcase__ = None
268         self.__set_base_data__(**kwargs)
269         self.__configure_logging__(module_names)
270
271     def __set_base_data__(self, **kwargs):
272         for key in set([key for key in self.DEFAULT_BASE_DATA.keys()] + [key for key in kwargs.
273             keys()]):
274             self[key] = kwargs.get(key, self.DEFAULT_BASE_DATA.get(key))
275             self[self.KEY_TESTCASELIST] = {}
276             self[self.KEY_UID_LIST] = []
277
278     def __configure_logging__(self, module_names):
279         #
280         # Configure logging for testSession
281         #
282         logging_config = dict(
283             version=1,
284             formatters={
285                 'short': {
286                     'format': SHORT_FMT,
287                 },
288                 'long': {
289                     'format': LONG_FMT,
290                 },
291             },
292             handlers={
293                 'console': {
294                     'level': 'DEBUG',
295                     'class': 'logging.NullHandler',
```

Unittest for report

```

295         'formatter': 'short ',
296     },
297     'module_logs': {
298         'level': 'DEBUG',
299         'class': 'report.collectingHandler',
300         'formatter': 'short ',
301     },
302     'testcase_logs': {
303         'level': 'DEBUG',
304         'class': 'report.collectingTestcaseHandler',
305         'formatter': 'short ',
306     },
307 },
308     loggers=self.__module_loggers__(module_names),
309 )
310 dictConfig(logging_config)
311
312 def __module_loggers__(self, module_names):
313     rv = {}
314     rv['__tLogger__'] = dict(handlers=['console', 'testcase_logs'], level='DEBUG', propagate=False)
315     for name in module_names + ['__mLogger__']:
316         rv[name] = dict(handlers=['console', 'module_logs'], level='DEBUG', propagate=False)
317     return rv
318
319 def testCase(self, name, testcase_execution_level, test_method, *args, **kwargs):
320     if testcase_execution_level <= self[self.KEY_EXEC_LVL]:
321         tLogger = logging.getLogger('__tLogger__')
322         tHandler = collectingTestcaseHandler()
323         if len(tHandler.MY_LOGS) > 0:
324             raise AttributeError("Testcaselogger shall be empty after closing testcase!")
325         tLogger._log(logging.DEBUG, name, None)
326         if len(tHandler.MY_LOGS) != 1:
327             raise AttributeError("Testcaselogger shall have only one entry for the main
328 testcase (temporary)!")
329         self.__testcase__ = tHandler.get_logs()[0]
330         test_method(logging.getLogger('__tLogger__'), *args, **kwargs)
331         self.__close_active_testcase__()
332
333 def __close_active_testcase__(self):
334     if self.__testcase__ is not None:
335         name = self.__testcase__.get('message')
336         #
337         # Add testcase
338         #
339         tch = collectingTestcaseHandler()
340         self.__testcase__[ 'testcaseLogger' ] = tch.get_logs()
341         if name in self[self.KEY_TESTCASELIST]:
342             raise AttributeError("Testcase named %s already exists" % name)
343         self[self.KEY_TESTCASELIST][name] = self.__testcase__
344         self[self.KEY_UID_LIST].append(name)
345         #
346         # Adapt testcase data
347         #
348         self[self.KEY_TESTCASELIST][name][ 'levelNo' ] = 0
349         self[self.KEY_TESTCASELIST][name][ 'time_consumption' ] = 0.
350         for teststep in self[self.KEY_TESTCASELIST][name][ 'testcaseLogger' ]:
351             # store maximum level to testcase
352             if teststep.get('levelNo') > self[self.KEY_TESTCASELIST][name][ 'levelNo' ]:
353                 self[self.KEY_TESTCASELIST][name][ 'levelNo' ] = teststep.get('levelNo')
354                 self[self.KEY_TESTCASELIST][name][ 'levelName' ] = teststep.get('levelName')
355             # store time_consumption for teststep

```

Unittest for report

```
355         try:
356             teststep['time_consumption'] = teststep['created'] - teststep['moduleLogger'
357             ][-1]['created']
358         except IndexError:
359             teststep['time_consumption'] = 0.
360             # Increment testcase time_consumption
361             # Increment testcase counters
362             #
363             self[self.KEY_ALL_TESTS] += 1
364             if self[self.KEY_TESTCASELIST][name]['levelno'] <= logging.INFO:
365                 self[self.KEY_SUCCESS_TESTS] += 1
366             elif self[self.KEY_TESTCASELIST][name]['levelno'] >= logging.ERROR:
367                 self[self.KEY_FAILED_TESTS] += 1
368             else:
369                 self[self.KEY_POSSIBLY_FAILED_TESTS] += 1
370             # Set testcase time and time_consumption
371             self[self.KEY_TESTCASELIST][name]['time_start'] = self.__testcase__['asctime']
372             self[self.KEY_TESTCASELIST][name]['time_finished'] = teststep['asctime']
373             self[self.KEY_TESTCASELIST][name]['time_consumption'] = teststep['created'] - self.
374             __testcase__['created']
375             # Set testcase time consumption
376             self['time_consumption'] += self[self.KEY_TESTCASELIST][name]['time_consumption']
377             self.__testcase__ = None
```