

Nliteme User Guide

Contents

1.	Overview.....	3
2.	Navigation bar.....	3
3.	Dashboard view.....	3
4.	List Views.....	4
5.	Details Views.....	5
6.	Build Compare view.....	6
7.	High level reports.....	7
7.1	Feature based high level report.....	7
7.2	Test Suite based high level report.....	8
7.3	Feature & Test Suite based high level report.....	8
7.4	High level report details.....	8
8.	Feature support.....	9
8.1	Setting coverage percentage.....	10
8.2	Feature specific high level reports.....	11
8.3	Feature related information in other views.....	11
9.	Defects support.....	12
9.1	Defect specific high level reports.....	13
10.	RESTful API for data base queries.....	15
10.1	Creating queries based on list views.....	15
10.1.1	Setting URL query action.....	15
10.1.2	Handling pagination.....	16
10.1.3	Handling sorting parameters and order.....	17
10.1.4	Setting query filters.....	19
10.2	Creating queries based on details views.....	21
10.3	Creating queries based on High Level Report views.....	22

1. Overview

This user guide has been prepared based on the nliteme deployment in the NR stack and phy development project. It presents the basic web interface views of the application.

2. Navigation bar

The top navigation bar is used to navigate to all Nliteme views. The following views are available:

- Dashboard – collection of latest test activities widgets grouped in different tabs
- Builds – list view for browsing the available daily builds records
- Build Increments – list view for browsing the available daily build increment records
- Test Results – list view for browsing the test execution results
- Test Cases - list view for browsing the test case definitions
- Test Suites - list view for browsing the test suite definitions
- Test Lines - list view for browsing the test line definitions
- Features - list view for browsing the feature definitions
- Build Compare – view for comparing daily builds results
- High Level Reports – drop down list with different reporting views

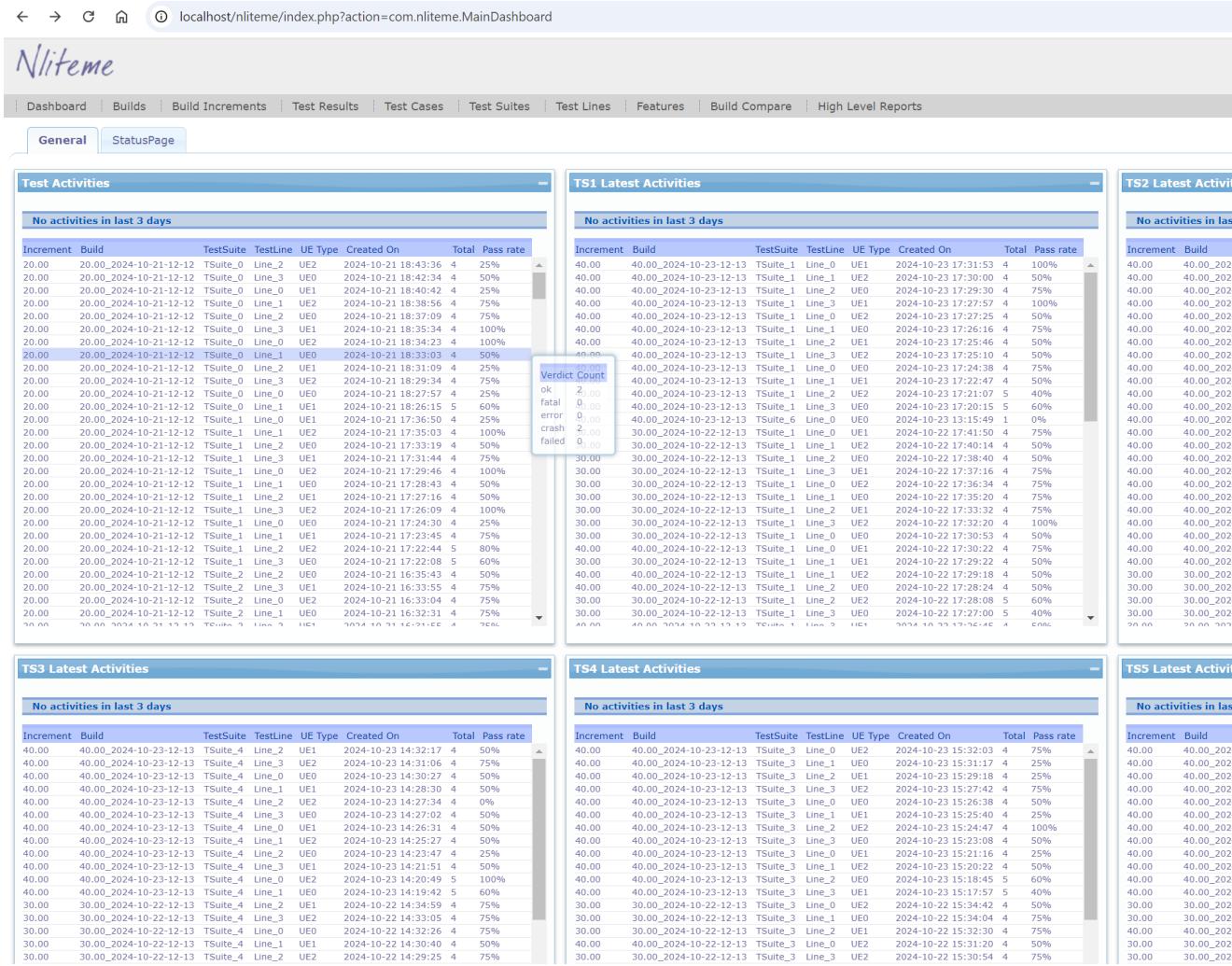
3. Dashboard view

The dashboard view summarized that latest test execution activities and consists of activity widgets grouped in different tabs. The dashboard configuration is described in the [NlitemeInstallationGuide.docx](#) chapter 6.

Each activity widget provides the test results summary with pass rate, total number of executed testcases and links to detailed results for latest builds grouped by daily build, test suite, test line and the UE type.

The tooltip with the number of test results per verdict type summary pops up when hovering on each activity summary line.

The activities are hyperlinked to the test results list views filtered accordingly.



4. List Views

The list views present information in the tabular form with pagination. Each page consists of maximum 100 table rows. The type of columns is list view specific e.g. different columns are presented in test results and builds list views. Below is an example of the test results view:

[localhost/nliteme/index.php?action=com.nliteme.MainTestResults&build=20.00_2024-10-21-12-12](#)

Test Results (300)

« Previous	1	2	3	Next »						
TestCase	Increment	Build	TestSuite	Verdict	TestLine	Feature	Feature Coverage %	UE Type	Defect	Created On
tc_0099	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_3	1004 Feature_4	5	UE0	N/A	2024-10-21 17:22:08
tc_0098	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_2	1004 Feature_4	5	UE2	N/A	2024-10-21 17:22:44
tc_0097	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_1	1004 Feature_4	5	UE1	N/A	2024-10-21 17:23:45
tc_0096	20.00	20.00_2024-10-21-12-12	TSuite_1	fatal	Line_0	1004 Feature_4	5	UE0	1700	2024-10-21 17:24:30
tc_0095	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_3	1004 Feature_4	5	UE2	N/A	2024-10-21 17:26:09
tc_0094	20.00	20.00_2024-10-21-12-12	TSuite_1	crash	Line_2	1004 Feature_4	5	UE1	1900	2024-10-21 17:27:16
tc_0093	20.00	20.00_2024-10-21-12-12	TSuite_1	crash	Line_1	1004 Feature_4	5	UE0	1900	2024-10-21 17:28:43
tc_0092	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_0	1004 Feature_4	5	UE2	N/A	2024-10-21 17:29:46
tc_0091	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_3	1004 Feature_4	5	UE1	N/A	2024-10-21 17:31:44
tc_0090	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_2	1004 Feature_4	5	UE0	N/A	2024-10-21 17:33:19
tc_009	20.00	20.00_2024-10-21-12-12	TSuite_0	✓	Line_1	1000 Feature_0	5	UE0	N/A	2024-10-21 19:20:11
tc_0089	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_1	1004 Feature_4	5	UE2	N/A	2024-10-21 17:35:03
tc_0088	20.00	20.00_2024-10-21-12-12	TSuite_1	error	Line_0	1004 Feature_4	5	UE1	1800	2024-10-21 17:36:50
tc_0087	20.00	20.00_2024-10-21-12-12	TSuite_1	crash	Line_3	1004 Feature_4	5	UE0	1900	2024-10-21 17:38:23
tc_0086	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_2	1004 Feature_4	5	UE2	N/A	2024-10-21 17:40:09
tc_0085	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_1	1004 Feature_4	5	UE1	N/A	2024-10-21 17:41:39

When hovering the cursor on the specific column the tooltip with column specific description e.g. build, test case description is shown e.g. if the cursor is pointed on the verdict column icon the test result specific description is shown in the tooltip:

[localhost/nliteme/index.php?action=com.nliteme.MainTestResults&build=20.00_2024-10-21-12-12](#)

Test Results (300)

« Previous	1	2	3	Next »				
TestCase	Increment	Build	TestSuite	Verdict	TestLine	Feature	Feature Coverage %	UE Type
tc_0099	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_3	1004 Feature_4	5	UE0
tc_0098	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_2	1004 Feature_4	5	UE2
tc_0097	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_1	1004 Feature_4	5	UE1
tc_0096	20.00	20.00_2024-10-21-12-12	TSuite_1	fatal			0	
tc_0095	20.00	20.00_2024-10-21-12-12	TSuite_1	✓			2	
tc_0094	20.00	20.00_2024-10-21-12-12	TSuite_1	crash			1	
tc_0093	20.00	20.00_2024-10-21-12-12	TSuite_1	crash			0	
tc_0092	20.00	20.00_2024-10-21-12-12	TSuite_1	✓			2	
tc_0091	20.00	20.00_2024-10-21-12-12	TSuite_1	✓			1	
tc_0090	20.00	20.00_2024-10-21-12-12	TSuite_1	✓			0	
tc_009	20.00	20.00_2024-10-21-12-12	TSuite_0	✓			0	
tc_0089	20.00	20.00_2024-10-21-12-12	TSuite_1	✓			2	
tc_0088	20.00	20.00_2024-10-21-12-12	TSuite_1	error	Line_0	1004 Feature_4	5	UE1
tc_0087	20.00	20.00_2024-10-21-12-12	TSuite_1	crash	Line_3	1004 Feature_4	5	UE0
tc_0086	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_2	1004 Feature_4	5	UE2
tc_0085	20.00	20.00_2024-10-21-12-12	TSuite_1	✓	Line_1	1004 Feature_4	5	UE1

Some tables contain quick link icons e.g. in the test results view links to the test results log file and opens a history of test results (for a given build increment, line, ue type)

The check box (left most column) allows selecting records, which can then be deleted using the trash bin icon in the top right current of the table. Caution. Clicking the delete icon removes the selected records from the database without requesting a confirmation!

localhost/nliteme/index.php?action=com.nliteme.MainTestResults&build=20.00_2024-10-21-12-12

Test Results (300)											
	TestCase	Increment	Build	TestSuite	Verdict	TestLine	Feature	Feature Coverage %	UE Type	Defect	Created On
<input checked="" type="checkbox"/>	tc_0099	20.00	20.00_2024-10-21-12-12	TSuite_1	Pass	Line_3	1004 Feature_4	<div style="width: 5%;">5</div>	UE0	N/A	2024-10-21 17:22:08
<input checked="" type="checkbox"/>	tc_0098	20.00	20.00_2024-10-21-12-12	TSuite_1	Pass	Line_2	1004 Feature_4	<div style="width: 5%;">5</div>	UE2	N/A	2024-10-21 17:22:44
<input type="checkbox"/>	tc_0097	20.00	20.00_2024-10-21-12-12	TSuite_1	Pass	Line_1	1004 Feature_4	<div style="width: 5%;">5</div>	UE1	N/A	2024-10-21 17:23:45
<input type="checkbox"/>	tc_0096	20.00	20.00_2024-10-21-12-12	TSuite_1	Fatal	Line_0	1004 Feature_4	<div style="width: 5%;">5</div>	UE0	1700	2024-10-21 17:24:30
<input type="checkbox"/>	tc_0095	20.00	20.00_2024-10-21-12-12	TSuite_1	Pass	Line_3	1004 Feature_4	<div style="width: 5%;">5</div>	UE2	N/A	2024-10-21 17:26:09
<input checked="" type="checkbox"/>	tc_0094	20.00	20.00_2024-10-21-12-12	TSuite_1	Pass	Line_2	1004 Feature_4	<div style="width: 5%;">5</div>	UE1	1900	2024-10-21 17:27:16
<input checked="" type="checkbox"/>	tc_0093	20.00	20.00_2024-10-21-12-12	TSuite_1	Pass	Line_1	1004 Feature_4	<div style="width: 5%;">5</div>	UE0	1900	2024-10-21 17:28:43
<input type="checkbox"/>	tc_0092	20.00	20.00_2024-10-21-12-12	TSuite_1	Pass	Line_0	1004 Feature_4	<div style="width: 5%;">5</div>	UE2	N/A	2024-10-21 17:29:46
<input type="checkbox"/>	tc_0091	20.00	20.00_2024-10-21-12-12	TSuite_1	Pass	Line_3	1004 Feature_4	<div style="width: 5%;">5</div>	UE1	N/A	2024-10-21 17:31:44
<input type="checkbox"/>	tc_0090	20.00	20.00_2024-10-21-12-12	TSuite_1	Pass	Line_2	1004 Feature_4	<div style="width: 5%;">5</div>	UE0	N/A	2024-10-21 17:33:19

5. Details Views

The columns in the list view contains the hyperlinks to the record specific details view e.g. clicking the record specific verdict icon in the test results list view redirects to the test result detail view:

Test Result: 3205

Details

Log File Path * ftp://ftp.logs.server/tc_0096_20241021_172430.7z

Created On *	2024-10-21 17:24:30	Increment *	20.00
Build *	20.00_2024-10-21-12-12	TestSuite *	TSuite_1
TestCase *	tc_0096	TestLine *	Line_0
UE Type *	UE0	Defect *	1700
Duration [s] *	45	Feature *	1004 Feature_4

Description

```
This is a multiline
description with special characters like:
';'=-0%)-!-
Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.
Stet clita kasd gubergren, no sea takinata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.
At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren,
no sea takinata sanctus est Lorem ipsum dolor sit amet.
```

whereas if the user click on the record specific build name it will be redirected to the build details view (and so on):

Build: 31

Details

Build *	40.00_2024-10-23-12-13	Tested On *	2024-10-23 19:05:33
Created On *	2024-10-23 13:15:49	Tested On *	2024-10-23 19:25:30
Increment *	40.00		

Related

Associated Features

High Level Reports for Builds

Increment	Build	Feature	Feature Coverage %	Created On	Total	Pass rate
40.00	40.00_2024-10-23-12-13	1000 Feature_0	<div style="width: 55%;">Passed: 55% Executed: 95% Defined: 95%</div>	2024-10-23 19:05:33	19	57.9%
40.00	40.00_2024-10-23-12-13	1001 Feature_1	<div style="width: 55%;">Passed: 55% Executed: 100% Defined: 100%</div>	2024-10-23 18:57:15	20	55%
40.00	40.00_2024-10-23-12-13	1002 Feature_2	<div style="width: 85%;">Passed: 85% Executed: 100% Defined: 100%</div>	2024-10-23 18:22:08	20	85%
40.00	40.00_2024-10-23-12-13	1003 Feature_3	<div style="width: 70%;">Passed: 70% Executed: 100% Defined: 100%</div>	2024-10-23 18:05:53	20	70%
40.00	40.00_2024-10-23-12-13	1004 Feature_4	<div style="width: 50%;">Passed: 50% Executed: 100% Defined: 100%</div>	2024-10-23 17:41:43	20	50%
40.00	40.00_2024-10-23-12-13	1005 Feature_5	<div style="width: 65%;">Passed: 65% Executed: 100% Defined: 100%</div>	2024-10-23 17:17:50	20	65%
40.00	40.00_2024-10-23-12-13	1006 Feature_6	<div style="width: 60%;">Passed: 60% Executed: 100% Defined: 100%</div>	2024-10-23 16:51:15	20	60%
40.00	40.00_2024-10-23-12-13	1007 Feature_7	<div style="width: 65%;">Passed: 65% Executed: 100% Defined: 100%</div>	2024-10-23 16:12:57	20	65%
40.00	40.00_2024-10-23-12-13	1008 Feature_8	<div style="width: 60%;">Passed: 60% Executed: 100% Defined: 100%</div>	2024-10-23 16:02:53	20	60%
40.00	40.00_2024-10-23-12-13	1009 Feature_9	<div style="width: 35%;">Passed: 35% Executed: 100% Defined: 100%</div>	2024-10-23 15:35:30	20	35%

6. Build Compare view

The build compare view is used for comparing test results from different daily builds. The view uses lazy loading technique i.e. it loads the new build compare records on page scrolling until all records are loaded

TestCase	TestLine	TestSuite	Feature	UE Type	40.00	30.00
tc_0050	Line_2	TSuite_1	1002 Feature_2	UE2	✗	✗
tc_0051	Line_3	TSuite_1	1002 Feature_2	UE0	✓	✓
tc_0052	Line_0	TSuite_1	1002 Feature_2	UE1	✓	✓
tc_0053	Line_1	TSuite_1	1002 Feature_2	UE2	✓	✗
tc_0054	Line_2	TSuite_1	1002 Feature_2	UE0	✓	✓
tc_0055	Line_3	TSuite_1	1002 Feature_2	UE1	✓	✓
tc_0056	Line_0	TSuite_1	1002 Feature_2	UE2	✓	✗
tc_0057	Line_1	TSuite_1	1002 Feature_2	UE0	✓	✓
tc_0058	Line_2	TSuite_1	1002 Feature_2	UE1	✓	✓
tc_0059	Line_3	TSuite_1	1002 Feature_2	UE2	✓	✓

The filtering form allows the selection of compare criteria such as daily build names, test cases, ue types, time frame etc. as well as a compare by option that could be verdict, pass rate percentage, test execution duration etc.

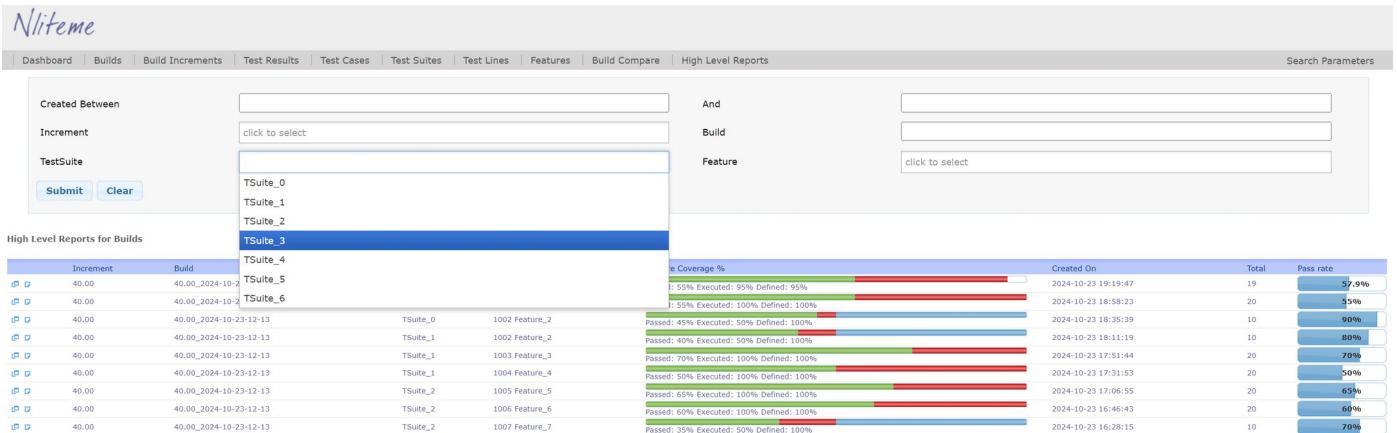
7. High level reports

The following high level reports are available:

- High Level Reports
- Features
- Testsuites
- Features & Testsuites
- Defects & Testcases

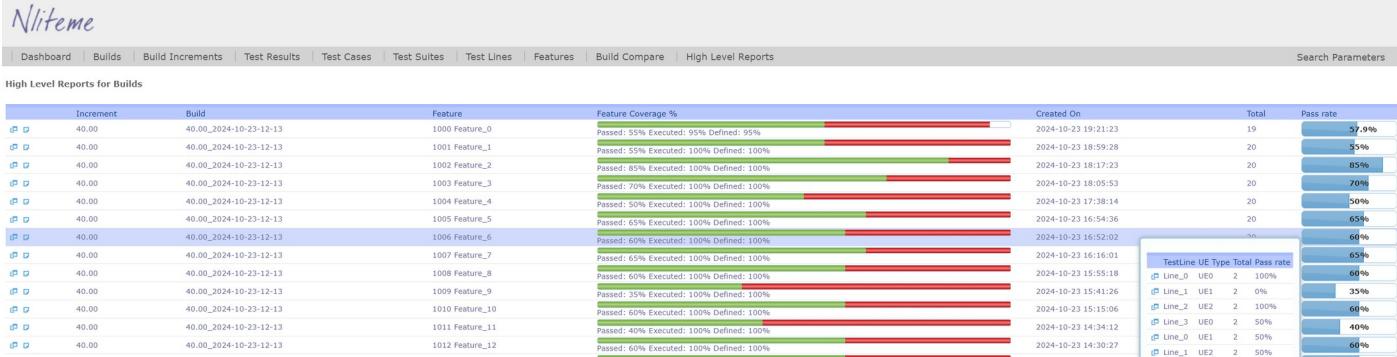
The Features, Testsuites, Feature & Testsuites reports present the test execution pass rate in form of lazy loaded tables (i.e. new records are loaded when the page is scrolled) grouped as the report name suggests. The pass rate criteria assume the certain test case is considered passed if at least one execution has a verdict OK (e.g. in case there are more execution of the same test case e.g. on different test lines and if one of them is OK, the test case is considered as passed).

The filtering form (opened via Search Parameters option) can be used to filter by addition criteria

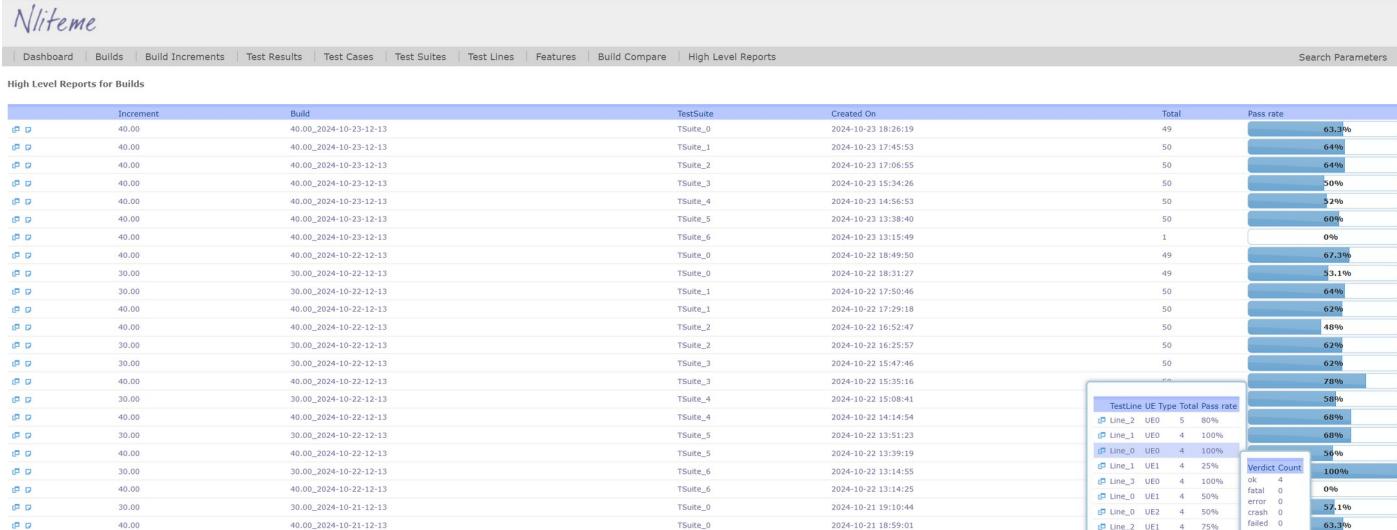


The quick link icons in each record redirect to the filtered test result list view and to the high level report details view .

7.1 Feature based high level report



7.2 Test Suite based high level report



7.3 Feature & Test Suite based high level report

High Level Reports for Builds										Search Parameters	
Increment	Build	TestSuite	Feature	Feature Coverage %			Created On	Total	Pass rate		
40.00	40.00_2024-10-23-12-13	TSuite_0	1000 Feature_0	Pas	55%	Exe	95%	95%	57.9%		
40.00	40.00_2024-10-23-12-13	TSuite_0	1001 Feature_1	Pas	55%	Exe	100%	100%	5%		
40.00	40.00_2024-10-23-12-13	TSuite_0	1002 Feature_2	Pas	45%	Exe	100%	100%	90%		
40.00	40.00_2024-10-23-12-13	TSuite_1	1002 Feature_2	Pas	40%	Exe	100%	100%	80%		
40.00	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Pas	70%	Exe	100%	100%	70%		
40.00	40.00_2024-10-23-12-13	TSuite_1	1004 Feature_4	Pas	50%	Exe	100%	100%	50%		
40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	65%	Exe	100%	100%	65%		
40.00	40.00_2024-10-23-12-13	TSuite_2	1006 Feature_6	Pas	60%	Exe	100%	100%	60%		
40.00	40.00_2024-10-23-12-13	TSuite_2	1007 Feature_7	Pas	35%	Exe	100%	100%	70%		
40.00	40.00_2024-10-23-12-13	TSuite_3	1007 Feature_7	Pas	30%	Exe	100%	100%	60%		
40.00	40.00_2024-10-23-12-13	TSuite_3	1008 Feature_8	Pas	60%	Exe	100%	100%	60%		
40.00	40.00_2024-10-23-12-13	TSuite_3	1009 Feature_9	Pas	35%	Exe	100%	100%	35%		
40.00	40.00_2024-10-23-12-13	TSuite_4	1010 Feature_10	Pas	60%	Exe	100%	100%	60%		
40.00	40.00_2024-10-23-12-13	TSuite_4	1011 Feature_11	Pas	40%	Exe	100%	100%	40%		

7.4 High level report details

Each record in the high level report list view has a link to the more detailed view i.e. high level report details.

Simply click on the  quick link icon or Pass rate column to open detail view, which consist of the pass rate per test line and test case verdict summary table widgets. The records in both tables contains hyperlink to the filtered test results list views.

High Level Report - Features & Testsuites										Search Parameters	
Pass Rate Per Test Line										Search Parameters	
TestLine	UE Type	Build	TestSuite	Feature	Feature Coverage %			Created On	Total	Pass rate	
Line_0	UE1	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	0%	10%	100%	2	0%	
Line_1	UE2	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	5%	10%	100%	2	50%	
Line_2	UE0	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	5%	10%	100%	2	50%	
Line_3	UE1	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	5%	10%	100%	2	50%	
Line_0	UE2	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	5%	10%	100%	2	50%	
Line_1	UE0	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	10%	10%	100%	2	100%	
Line_0	UE0	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	5%	10%	100%	2	100%	
Line_1	UE1	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	5%	10%	100%	1	100%	
Line_2	UE2	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	5%	10%	100%	1	100%	
Line_3	UE0	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	5%	10%	100%	1	100%	
Line_2	UE1	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	5%	10%	100%	2	50%	
Line_3	UE2	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Pas	10%	10%	100%	2	100%	
Test Cases Verdict											
High Level Report - Test Cases Verdict Summary (20)											
TestCase	Increment	Build	TestSuite	Feature	Feature Coverage %			Created On	Total	Pass rate	Verdict
tc_00100	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 17:19:10	2	0%	
tc_00101	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 17:17:50	2	50%	
tc_00102	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 17:17:08	2	50%	
tc_00103	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 17:15:34	2	50%	
tc_00104	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 17:13:45	2	50%	
tc_00105	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 17:12:28	2	100%	
tc_00106	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 17:08:04	1	100%	
tc_00107	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 17:06:55	1	100%	
tc_00108	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 17:05:51	1	100%	
tc_00109	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 17:04:26	1	100%	
tc_00110	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 16:54:36	2	50%	
tc_00111	40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5			5	2024-10-23 16:53:26	2	100%	

8. Feature support

The Feature view added in the nliteme version 1.2 provides an option to map the test cases to a particular feature (which can also be defined in the external feature management tool) and specify the percentage of the feature coverage by the particular test cases.

Feature definitions are stored in a separate table in the database. A Feature definition consists of:

- a feature name (it can also be just a number of the defect) - has to be unique, mandatory
- a record id (unique identifier auto-generated by the database),
- a hyperlink to the feature defined outside of nliteme (e.g. in Jazz) - optional
- a description – optional

Features can be listed and filtered via link available in the navigation bar:

Features (17)				
	Feature	Created On	Num of Testcases	Feature Coverage %
<input type="checkbox"/>	1000 Feature_0	2024-10-20 19:02:02	19	<div style="width: 95%;">95</div>
<input type="checkbox"/>	1001 Feature_1	2024-10-20 18:35:45	20	<div style="width: 100%;">100</div>
<input type="checkbox"/>	1002 Feature_2	2024-10-20 18:11:55	20	<div style="width: 100%;">100</div>
<input type="checkbox"/>	1003 Feature_3	2024-10-20 17:47:42	20	<div style="width: 100%;">100</div>
<input type="checkbox"/>	1004 Feature_4	2024-10-20 17:22:20	20	<div style="width: 100%;">100</div>
<input type="checkbox"/>	1005 Feature_5	2024-10-20 16:56:27	20	<div style="width: 100%;">100</div>
<input type="checkbox"/>	1006 Feature_6	2024-10-20 16:29:58	20	<div style="width: 100%;">100</div>

For each defined feature the total number of associated test cases as well as the total feature coverage percentage is shown.

NOTE 1. The feature coverage is an aggregated value of feature test coverage defined for each associated test case and can exceed the 100%. This is to allow the user to select a subset of test cases associated with a given feature that is considered to fully cover the feature for testing with the particular SW build.

NOTE 2. There is a default feature with the name: *Unknown*. All the test cases that do not have an associated feature are assigned to *Unknown* feature.

8.1 Setting coverage percentage

The feature test coverage is defined for each test case in the Test Case details view (from the navigation bar Test Cases, then select the Test Case name to open the details view:

Test Cases (19)				
	TestCase	Feature	Feature Coverage %	Created On
<input type="checkbox"/>	tc_001	1000 Feature_0	<div style="width: 5%;">5</div>	2024-10-20 19:23:40
<input checked="" type="checkbox"/>	tc_002	1000 Feature_0	<div style="width: 5%;">5</div>	2024-10-20 19:22:22
<input type="checkbox"/>	tc_003	1000 Feature_0	<div style="width: 5%;">5</div>	2024-10-20 19:21:26
<input type="checkbox"/>	tc_004	1000 Feature_0	<div style="width: 5%;">5</div>	2024-10-20 19:20:07

Test case: tc_002

Details
TestCase * <input type="text" value="tc_002"/>
Feature * <input type="text" value="1000 Feature_0"/>
Created On * <input type="text" value="2024-10-20 19:22:22"/>
Feature Coverage % * <input type="text" value="5"/>
Description
This is a test case description Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est lorem ipsum dolor sit amet.

8.2 Feature specific high level reports

The high level report also mentioned in sec. 7.1 and sec. 7.3 contains the feature coverage multicolor bar built from the following colored bar stacked on each other:

- Blue bar - the coverage of the defined test cases (bottom),
- Red bar - the coverage of the executed test cases (middle),
- Green bar - the coverage of the passed test case executions (top):

Nliteme

Dashboard Builds Build Increments Test Results Test Cases Test Suites Test Lines Features Build Compare High Level Reports Search Parameters

High Level Reports for Builds

Increment	Build	TestSuite	Feature	Feature Coverage %	Created On	Total	Pass rate
40.00	40.00_2024-10-23-12-13	TSuite_0	1000 Feature_0	Passed: 55% Executed: 95% Defined: 95%	2024-10-23 19:19:47	19	57.9%
40.00	40.00_2024-10-23-12-13	TSuite_0	1001 Feature_1	Passed: 55% Executed: 100% Defined: 100%	2024-10-23 18:58:23	20	55%
40.00	40.00_2024-10-23-12-13	TSuite_0	1002 Feature_2	Passed: 45% Executed: 50% Defined: 100%	2024-10-23 18:23:37	20	90%
40.00	40.00_2024-10-23-12-13	TSuite_1	1002 Feature_2	Passed: 40% Executed: 50% Defined: 100%	2024-10-23 18:12:52	10	80%
40.00	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 70% Executed: 100% Defined: 100%	2024-10-23 17:52:42	20	70%
40.00	40.00_2024-10-23-12-13	TSuite_1	1004 Feature_4	Passed: 50% Executed: 100% Defined: 100%	2024-10-23 17:32:46	20	50%
40.00	40.00_2024-10-23-12-13	TSuite_2	1005 Feature_5	Passed: 65% Executed: 100% Defined: 100%	2024-10-23 16:53:26	20	65%
40.00	40.00_2024-10-23-12-13	TSuite_2	1006 Feature_6	Passed: 60% Executed: 100% Defined: 100%	2024-10-23 16:36:41	20	60%
40.00	40.00_2024-10-23-12-13	TSuite_2	1007 Feature_7	Passed: 35% Executed: 50% Defined: 100%	2024-10-23 16:25:48	10	70%

After clicking the feature name the user is redirected to the Feature details view. However, in case external hyper link (Ext URL) is defined for the given feature, the link is used as a landing page (e.g. Jazz page is opened).

The detailed high level report with pass rate per test line or particular test case for a given build and feature is available by clicking on the Feature Coverage bar, Created on, Total or Pass rate column:

Nliteme

Dashboard Builds Build Increments Test Results Test Cases Test Suites Test Lines Features Build Compare High Level Reports

High Level Report - Features & Testsuites

Pass Rate Per Test Line

TestLine	UE Type	Build	TestSuite	Feature	Feature Coverage %	Created On	Total	Pass rate
Line_0	UE0	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 10% Defined: 100%	2024-10-23 18:05:53	2	50%
Line_1	UE1	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 10% Defined: 100%	2024-10-23 18:05:02	2	50%
Line_2	UE2	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 10% Defined: 100%	2024-10-23 18:03:23	2	50%
Line_3	UE0	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 10% Defined: 100%	2024-10-23 18:02:34	2	50%
Line_0	UE1	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 10% Executed: 10% Defined: 100%	2024-10-23 18:01:18	2	100%
Line_0	UE2	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 5% Defined: 100%	2024-10-23 17:56:43	1	100%
Line_1	UE0	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 5% Defined: 100%	2024-10-23 17:56:13	1	100%
Line_2	UE1	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 5% Defined: 100%	2024-10-23 17:54:38	1	100%
Line_3	UE2	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 10% Defined: 100%	2024-10-23 17:52:42	1	100%
Line_1	UE2	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 10% Defined: 100%	2024-10-23 17:45:53	2	50%
Line_2	UE0	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 10% Defined: 100%	2024-10-23 17:44:19	2	50%
Line_3	UE1	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 10% Executed: 10% Defined: 100%	2024-10-23 17:42:51	2	100%

Test Cases Verdict

High Level Report - Test Cases Verdict Summary (20)

TestCase	Increment	Build	TestSuite	Feature	Feature Coverage %	Created On	Total	Pass rate	Verdict
tc_0060	40.00	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 10% Defined: 100%	2024-10-23 18:05:53	2	50%	✗
tc_0061	40.00	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 10% Defined: 100%	2024-10-23 18:05:02	2	50%	✓
tc_0062	40.00	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 5% Defined: 100%	2024-10-23 18:03:23	2	50%	✗
tc_0063	40.00	40.00_2024-10-23-12-13	TSuite_1	1003 Feature_3	Passed: 5% Executed: 5% Defined: 100%	2024-10-23 18:02:34	1	100%	✓

8.3 Feature related information in other views

The Build details view contains now an additional summary of the tested features:

Nliteme

Dashboard Builds Build Increments Test Results Test Cases Test Suites Test Lines Features Build Compare High Level Reports Search Parameters

Build: 31

Details

Build * 40.00_2024-10-23-12-13

Created On * 2024-10-23 13:15:49

Increment * 40.00

Related

Associated Features

Nliteme

High Level Reports for Builds

Increment	Build	Feature	Feature Coverage %	Created On	Total	Pass rate
40.00	40.00_2024-10-23-12-13	1000 Feature_0	Passed: 55% Executed: 95% Defined: 95%	2024-10-23 19:23:08	19	57.9%
40.00	40.00_2024-10-23-12-13	1001 Feature_1	Passed: 55% Executed: 100% Defined: 100%	2024-10-23 18:52:21	20	55%
40.00	40.00_2024-10-23-12-13	1002 Feature_2	Passed: 85% Executed: 100% Defined: 100%	2024-10-23 18:08:01	20	85%
40.00	40.00_2024-10-23-12-13	1003 Feature_3	Passed: 70% Executed: 100% Defined: 100%	2024-10-23 18:05:02	20	70%
40.00	40.00_2024-10-23-12-13	1004 Feature_4	Passed: 50% Executed: 100% Defined: 100%	2024-10-23 17:20:15	20	50%
40.00	40.00_2024-10-23-12-13	1005 Feature_5	Passed: 65% Executed: 100% Defined: 100%	2024-10-23 17:17:08	20	65%
40.00	40.00_2024-10-23-12-13	1006 Feature_6	Passed: 60% Executed: 100% Defined: 100%	2024-10-23 16:40:31	20	60%
40.00	40.00_2024-10-23-12-13	1007 Feature_7	Passed: 65% Executed: 100% Defined: 100%	2024-10-23 16:29:55	20	65%
40.00	40.00_2024-10-23-12-13	1008 Feature_8	Passed: 60% Executed: 100% Defined: 100%	2024-10-23 16:16:01	20	60%
40.00	40.00_2024-10-23-12-13	1009 Feature_9	Passed: 35% Executed: 100% Defined: 100%	2024-10-23 15:23:08	20	35%
40.00	40.00_2024-10-23-12-13	1010 Feature_10	Passed: 60% Executed: 100% Defined: 100%	2024-10-23 15:15:06	20	60%
40.00	40.00_2024-10-23-12-13	1011 Feature_11	Passed: 40% Executed: 100% Defined: 100%	2024-10-23 14:45:15	20	40%
40.00	40.00_2024-10-23-12-13	1012 Feature_12	Passed: 50% Executed: 100% Defined: 100%	2024-10-23 14:07:36	20	60%
40.00	40.00_2024-10-23-12-13	1013 Feature_13	Passed: 60% Executed: 100% Defined: 100%	2024-10-23 14:04:06	20	60%
40.00	40.00_2024-10-23-12-13	1014 Feature_14	Passed: 60% Executed: 100% Defined: 100%	2024-10-23 13:27:12	20	60%
40.00	40.00_2024-10-23-12-13	1015 Feature_15	Passed: 0% Executed: 5% Defined: 5%	2024-10-23 13:15:49	1	0%

The Test results list view table contains the Feature column showing the feature associated with the particular test execution result:

	TestCase	Increment	Build	TestSuite	Verdict	TestLine	Feature	Feature Coverage %	UE Type	Defect	Created On
<input type="checkbox"/>	tc_00248	40.00	40.00_2024-10-23-12-13	TSuite_4	✓	Line_0	1012 Feature_12	5	UE2	N/A	2024-10-23 14:20:49
<input type="checkbox"/>	tc_00236	40.00	40.00_2024-10-23-12-13	TSuite_4	✓	Line_0	1011 Feature_11	5	UE2	N/A	2024-10-23 14:34:12
<input type="checkbox"/>	tc_00224	40.00	40.00_2024-10-23-12-13	TSuite_4	✓	Line_0	1011 Feature_11	5	UE2	N/A	2024-10-23 14:49:27
<input type="checkbox"/>	tc_00212	40.00	40.00_2024-10-23-12-13	TSuite_4	✓	Line_0	1010 Feature_10	5	UE2	N/A	2024-10-23 15:04:23
<input type="checkbox"/>	tc_00200	40.00	40.00_2024-10-23-12-13	TSuite_4	✓	Line_0	1010 Feature_10	5	UE2	N/A	2024-10-23 15:16:45

9. Defects support

The nliteme ver. 1.3. introduces the option to map the work item number/id (work items aka defects are hosted externally, using dedicated defect tracking tools) with the test execution result. The mapped defects are shown in the test results list and test result details views. It is also possible to filter the views by the defect number.

Here is the test results list view example with the filtering options:

Test Results (56)

	TestCase	Increment	Build	TestSuite	Verdict	TestLine	Feature	Feature Coverage %	UE Type	Defect	Created On
<input type="checkbox"/>	tc_00229	40.00	40.00_2024-10-23-12-13	TSuite_4	fatal	Line_1	1011 Feature_11	5	UE1	1600	2024-10-23 14:43:15
<input type="checkbox"/>	tc_00213	30.00	30.00_2024-10-22-12-13	TSuite_4	fatal	Line_1	1010 Feature_10	5	UE0	1600	2024-10-22 15:04:30
<input type="checkbox"/>	tc_00225	30.00	30.00_2024-10-22-12-13	TSuite_4	failed	Line_1	1011 Feature_11	5	UE0	1600	2024-10-22 14:49:40
<input type="checkbox"/>	tc_00221	40.00	40.00_2024-10-22-12-13	TSuite_4	failed	Line_1	1011 Feature_11	5	UE2	1600	2024-10-22 14:49:05
<input type="checkbox"/>	tc_00233	30.00	30.00_2024-10-22-12-13	TSuite_4	failed	Line_1	1011 Feature_11	5	UE2	1600	2024-10-22 14:40:41
<input type="checkbox"/>	tc_00249	30.00	30.00_2024-10-22-12-13	TSuite_4	fatal	Line_1	1012 Feature_12	5	UE0	1600	2024-10-22 14:21:34
<input type="checkbox"/>	tc_00221	20.00	20.00_2024-10-21-12-12	TSuite_4	fatal	Line_1	1011 Feature_11	5	UE2	1600	2024-10-21 14:47:05

Note. the defect number may be hyperlinked with the defect tracking tool, therefore it is possible to directly navigate to defect details page just by clicking on the link.

In case there is no defect is mapped to the test execution result (e.g. because the test result is OK), the Defect column shows N/A e.g.

	TestCase	Increment	Build	TestSuite	Verdict	TestLine	Feature	Feature Coverage %	UE Type	Defect	Created On
<input type="checkbox"/>	tc_00243	40.00	40.00_2024-10-23-12-13	TSuite_4	✓	Line_3	1012 Feature_12	5	UE0	N/A	2024-10-23 14:27:02
<input type="checkbox"/>	tc_00231	40.00	40.00_2024-10-23-12-13	TSuite_4	fatal	Line_3	1011 Feature_11	5	UE0	1700	2024-10-23 14:40:51
<input type="checkbox"/>	tc_00219	40.00	40.00_2024-10-23-12-13	TSuite_4	✓	Line_3	1010 Feature_10	5	UE0	N/A	2024-10-23 14:55:10
<input type="checkbox"/>	tc_00207	40.00	40.00_2024-10-23-12-13	TSuite_4	fatal	Line_3	1010 Feature_10	5	UE0	1700	2024-10-23 15:09:22

9.1 Defect specific high level reports

The new report available from the navigation bar *High Level Reports->Defects & Testcases* summarizes the test results of builds by grouping the failing test cases to the known defects.



The report presents the total amount of test cases that failed when testing the build due to known defects in a lazy loaded table form. The newly failed test cases, not yet associated with any defect, are summarized in the record with the defect marked as N/A.

This view, just like other, provides the filtering options for best matching the needed information.

Note. Since only the failed test cases are included in this report, the pass rate column always shows 0%. After hovering on this column, the tooltip with failed test cases per test line is shown

Increment	Build	Defect	Created On	Total	Pass rate
40.00	40.00_2024-10-23-12-13	1800	2024-10-23 17:49:58	25	0%
40.00	40.00_2024-10-23-12-13	1900	2024-10-23 17:40:27	38	0%
40.00	40.00_2024-10-23-12-13	1600	2024-10-23 14:52:20	29	0%
40.00	40.00_2024-10-23-12-13	1700	2024-10-23 14:47:29	32	0%
30.00	30.00_2024-10-22-12-13	1600	2024-10-22 19:08:31	29	0%
30.00	30.00_2024-10-22-12-13	1800	2024-10-22 17:35:20	29	0%
40.00	40.00_2024-10-22-12-13	1700	2024-10-22 17:05:42	29	0%
40.00	40.00_2024-10-22-12-13	1600	2024-10-22 15:01:13	29	0%
40.00	40.00_2024-10-22-12-13	1800	2024-10-22 14:53:59	29	0%
30.00	30.00_2024-10-22-12-13	1900	2024-10-22 14:00:23	29	0%
30.00	30.00_2024-10-22-12-13	1700	2024-10-22 13:52:29	29	0%
40.00	40.00_2024-10-22-12-13	1900	2024-10-22 13:57:08	29	0%

Testline	UE Type	Total	Pass rate
Line_3	UE2	2	0%
Line_3	UE1	3	0%
Line_1	UE1	1	0%
Line_3	UE0	2	0%
Line_2	UE0	2	0%

Like in other high level report view, the quick link icons in each record redirect to the filtered test result list view and to the high level report details view .

The high level report details view consists of widgets summarizing the number of failed test cases per affected test lines as well as list of test cases failing due to the given defect.

TestLine	UE Type	Build	Defect	Created On	Total
Line_3	UE2	30.00_2024-10-22-12-13	1800	2024-10-22 19:30:55	2
Line_3	UE1	30.00_2024-10-22-12-13	1800	2024-10-22 19:05:34	3
Line_1	UE1	30.00_2024-10-22-12-13	1800	2024-10-22 18:21:27	1
Line_3	UE0	30.00_2024-10-22-12-13	1800	2024-10-22 18:14:19	2
Line_2	UE0	30.00_2024-10-22-12-13	1800	2024-10-22 18:11:28	2
Line_0	UE0	30.00_2024-10-22-12-13	1800	2024-10-22 18:04:21	2
Line_1	UE0	30.00_2024-10-22-12-13	1800	2024-10-22 17:35:20	2
Line_2	UE2	30.00_2024-10-22-12-13	1800	2024-10-22 16:41:05	3
Line_1	UE2	30.00_2024-10-22-12-13	1800	2024-10-22 16:21:57	3
Line_0	UE1	30.00_2024-10-22-12-13	1800	2024-10-22 16:09:19	2

TestCase	Increment	Build	TestSuite	Feature	Feature Coverage %	Defect
tc_0011	30.00	30.00_2024-10-22-12-13	TSuite_0	1006 Feature_0	5	1800
tc_00134	30.00	30.00_2024-10-22-12-13	TSuite_2	1006 Feature_6	5	1800
tc_00141	30.00	30.00_2024-10-22-12-13	TSuite_2	1007 Feature_7	5	1800
tc_00149	30.00	30.00_2024-10-22-12-13	TSuite_2	1007 Feature_7	5	1800
tc_00160	30.00	30.00_2024-10-22-12-13	TSuite_3	1008 Feature_8	5	1800
tc_00163	30.00	30.00_2024-10-22-12-13	TSuite_3	1008 Feature_8	5	1800
tc_00168	30.00	30.00_2024-10-22-12-13	TSuite_3	1008 Feature_8	5	1800
tc_00209	30.00	30.00_2024-10-22-12-13	TSuite_4	1010 Feature_10	5	1800
tc_00223	30.00	30.00_2024-10-22-12-13	TSuite_0	1001 Feature_3	5	1800
tc_00243	30.00	30.00_2024-10-22-12-13	TSuite_4	1012 Feature_12	5	1800
tc_00245	30.00	30.00_2024-10-22-12-13	TSuite_4	1012 Feature_12	5	1800
tc_00254	30.00	30.00_2024-10-22-12-13	TSuite_5	1012 Feature_12	5	1800
tc_00294	30.00	30.00_2024-10-22-12-13	TSuite_5	1014 Feature_14	5	1800
tc_00321	30.00	30.00_2024-10-22-12-13	TSuite_0	1001 Feature_1	5	1800
tc_0040	30.00	30.00_2024-10-22-12-13	TSuite_0	1002 Feature_2	5	1800

10. RESTful API for data base queries

Since the version 1.3 the nliteme allows to programmatically query the nliteme data base using the HTTP GET URL queries. The nliteme server parses the URL query parameters and queries the database. The result is returned in a form of json encoded HTTP response.

This functionality extends the usability of nliteme by other tools, allowing an easy and fast access to the database as well as building advanced applications for example new client based UIs, statistical and KPI tools, management and planning tools etc.

10.1 Creating queries based on list views

10.1.1 Setting URL query action

Depending on what type of information shall be retrieved from the database e.g. builds, test cases, test results, high level reports etc. the HTTP query URL has to specify the *action* parameter which is always prefixed with *com.nliteme.* string followed by the information type.

For example to query the test results specific information the *action* parameter would be:

```
action=com.nliteme.TestResults
```

whereas in case of builds specific queries:

```
action=com.nliteme.Builds
```

or in case of test suites high level report it would be:

```
action=com.nliteme.HighLevelReport&groupby[]={tsname}
```

Note. in this case also additional groupby[]={tsname} parameter has to be specified

The easiest way to learn what action has to be specified is to open the web application views and check the generated URL in the browser. The *action* parameter with the stripped *Main* prefix is in most cases the one to use e.g.

The screenshot shows a web browser window with the URL `localhost/nliteme/index.php?action=com.nliteme.MainTestCases` in the address bar. The URL is highlighted with a yellow oval. Below the browser, there is a screenshot of the nliteme web interface. The interface has a header with the nliteme logo and a navigation menu with links like Dashboard, Builds, Build Increments, Test Results, Test Cases, Test Suites, and Test Lines. The main content area is titled "Test Cases (300)" and shows a table with two rows. The table has columns for a checkbox, TestCase name, and Feature name. The first row has a checkbox next to "TestCase", the name "tc_0099", and the feature "1004 Feature_4". The second row has a checkbox next to "tc_0099", the name "tc_0099", and the feature "1004 Feature_4". There are navigation links "« Previous", "1", "2", "3", and "Next »" above the table.

would be converted to:

```
http://mu-prg-server0/nliteme/index.php?action=com.nliteme.TestCases
```

To inform the server that the response shall be encoded as json (and not plain html) the *usejson* parameter shall be included in the query URL i.e. *usejson* or *usejson=1*

The URL from the previous example would then look as follows:

```
http://mu-prg-server0/nliteme/index.php?action=com.nliteme.TestCases&usejson
```

The response would look like:

localhost/nliteme/index.php?action=com.nliteme.TestCases&usejson

```
{
  "content": {
    "detailActions": { ... }, // 7 items
    "fieldActions": { ... }, // 1 item
    "checkboxid": "tcid",
    "label": "TPL_TESTCASES",
    "header": { ... }, // 4 items
    "submitAction": "com.nliteme.TestCases",
    "body": [
      "0": {
        "tcid": "202",
        "createdate": "2024-10-20 17:22:20",
        "tcname": "tc_0099",
        "shortdescription": null,
        "description": null,
        "fid": "13",
        "coverage": "5",
        "fname": "1004 Feature_4",
        "hlink": "http://localhost/feature/testfeature?id=1004"
      },
      "1": {
        "tcid": "203",
        "createdate": "2024-10-20 17:24:03",
        "tcname": "tc_0098",
        "shortdescription": null,
        "description": null,
        "fid": "13",
        "coverage": "5",
        "fname": "1004 Feature_4",
        "hlink": "http://localhost/feature/testfeature?id=1004"
      },
      "2": {
        "tcid": "204",
        "createdate": "2024-10-20 17:25:51",
        "tcname": "tc_0097",
        "shortdescription": null,
        "description": null
      }
    ]
  }
}
```

The queried data are included in the structure identified by the json key: *body*.

10.1.2 Handling pagination

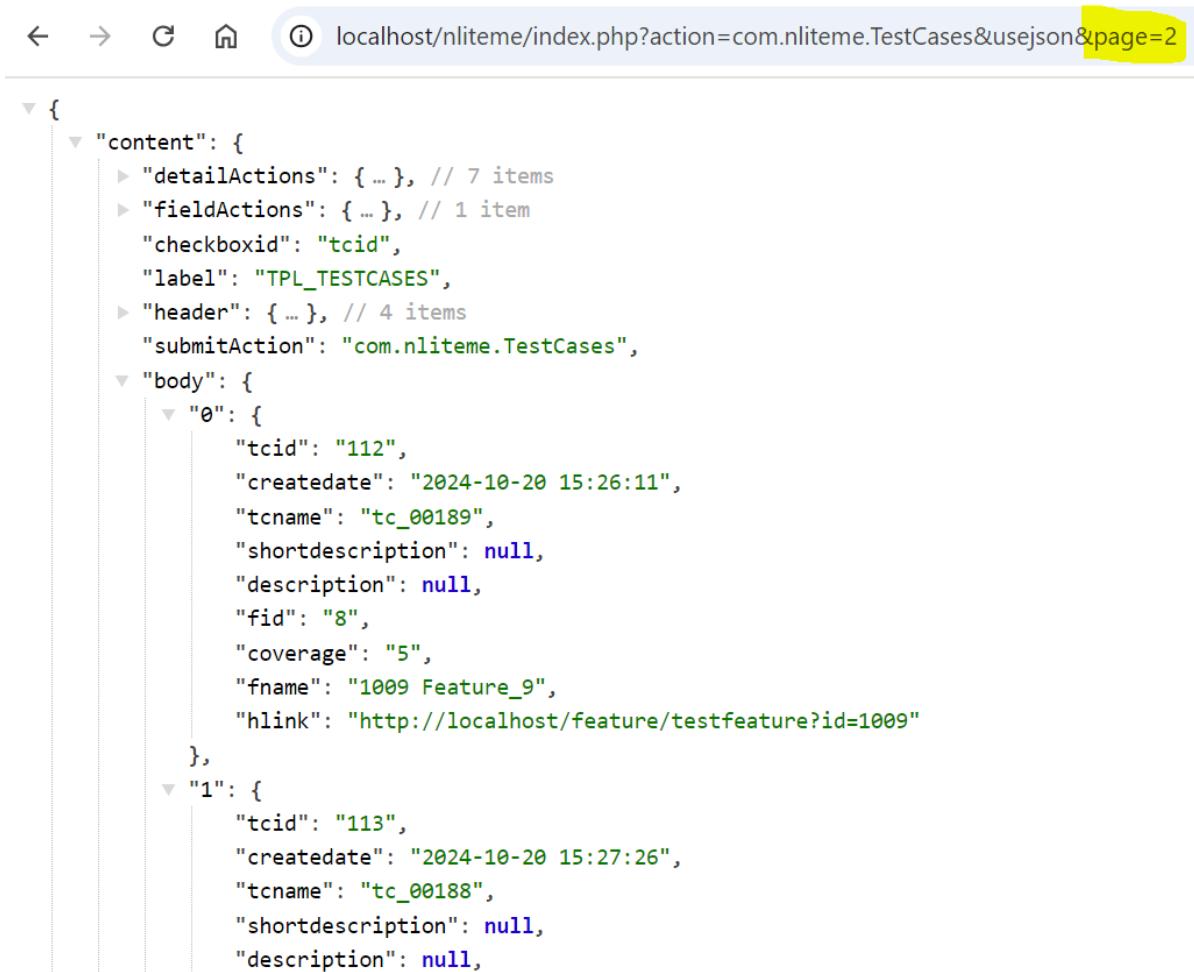
The list views use pagination with table size per page of max 100, therefore the single json response might not contain the complete data base query output. The total size of the database query response can be found by inspecting the *numberofrecords*:

localhost/nliteme/index.php?action=com.nliteme.TestCases&usejson

```
{
  "content": {
    "detailActions": { ... }, // 7 items
    "fieldActions": { ... }, // 1 item
    "checkboxid": "tcid",
    "label": "TPL_TESTCASES",
    "header": { ... }, // 4 items
    "submitAction": "com.nliteme.TestCases",
    "body": { ... }, // 100 items
    "numberofrecords": "300",
    "footer": {}
  }
}
```

The different pages requested by specifying the *page* number in the query URL:

<http://mu-prg-server0/nliteme/index.php?action=com.nliteme.TestCases&usejson&page=10>



The screenshot shows a browser window with the URL `localhost/nliteme/index.php?action=com.nliteme.TestCases&usejson&page=2`. The page displays a JSON object representing test cases. The structure includes a "content" section with various actions and a "body" section containing two test case entries, indexed 0 and 1.

```
{  
  "content": {  
    "detailActions": { ... }, // 7 items  
    "fieldActions": { ... }, // 1 item  
    "checkboxid": "tcid",  
    "label": "TPL_TESTCASES",  
    "header": { ... }, // 4 items  
    "submitAction": "com.nliteme.TestCases",  
  },  
  "body": {  
    "0": {  
      "tcid": "112",  
      "createdate": "2024-10-20 15:26:11",  
      "tcname": "tc_00189",  
      "shortdescription": null,  
      "description": null,  
      "fid": "8",  
      "coverage": "5",  
      "fname": "1009 Feature_9",  
      "hlink": "http://localhost/feature/testfeature?id=1009"  
    },  
    "1": {  
      "tcid": "113",  
      "createdate": "2024-10-20 15:27:26",  
      "tcname": "tc_00188",  
      "shortdescription": null,  
      "description": null,  
    }  
  }  
}
```

10.1.3 Handling sorting parameters and order

The queried data can also be sorted by different fields. For example, to sort the test cases by the test case name in the ascending order the query from the previous example would look like:

```
http://mu-prg-server0/nliteme/index.php?  
action=com.nliteme.TestCases&usejson&page=1&sort=tcname&order=ASC
```

localhost/nliteme/index.php?action=com.nliteme.TestCases&usejson&page=2&sort=tcname&order=ASC

```
▼ {
  "content": {
    ▶ "detailActions": { ... }, // 7 items
    ▶ "fieldActions": { ... }, // 1 item
    "checkboxid": "tcid",
    "label": "TPL_TESTCASES",
    ▶ "header": { ... }, // 4 items
    "submitAction": "com.nliteme.TestCases",
    ▶ "body": {
      ▶ "0": {
        "tcid": "273",
        "createdate": "2024-10-20 18:48:44",
        "tcname": "tc_00228",
        "shortdescription": null,
        "description": null,
        "fid": "16",
        "coverage": "5",
        "fname": "1001 Feature_1",
        "hlink": "http://localhost/feature/testfeature?id=1001"
      },
      ▶ "1": {
        "tcid": "21",
        "createdate": "2024-10-20 13:36:01",
        "tcname": "tc_00280",
        "shortdescription": null,
        "description": null,
        "fid": "3",
        "coverage": "5",
        "fname": "1014 Feature_14",
        "hlink": "http://localhost/feature/testfeature?id=1014"
      },
      ▶ "2": {
        "tcid": "20",
        "createdate": "2024-10-20 13:34:59",
        "tcname": "tc_00281",
      }
    }
  }
}
```

Or to sort by creation date starting from the latest:

http://mu-prg-server0/nliteme/index.php?
action=com.nliteme.TestCases&usejson&page=0&sort=createdate&order=DESC

```
◀ → ⌂ ⌄ localhost/nliteme/index.php?action=com.nliteme.TestCases&usejson&page=2&sort=tcname&order=DESC
```

```
▼ {
  "content": {
    ▶ "detailActions": { ... }, // 7 items
    ▶ "fieldActions": { ... }, // 1 item
    "checkboxid": "tcid",
    "label": "TPL_TESTCASES",
    ▶ "header": { ... }, // 4 items
    "submitAction": "com.nliteme.TestCases",
    ▶ "body": {
      ▶ "0": {
        "tcid": "112",
        "createdate": "2024-10-20 15:26:11",
        "tcname": "tc_00189",
        "shortdescription": null,
        "description": null,
        "fid": "8",
        "coverage": "5",
        "fname": "1009 Feature_9",
        "hlink": "http://localhost/feature/testfeature?id=1009"
      },
      ▶ "1": {
        "tcid": "113",
        "createdate": "2024-10-20 15:27:26",
        "tcname": "tc_00188",
        "shortdescription": null,
        "description": null,
        "fid": "8",
        "coverage": "5",
        "fname": "1009 Feature_9",
        "hlink": "http://localhost/feature/testfeature?id=1009"
      },
      ▶ "2": {
        "tcid": "114",
        "createdate": "2024-10-20 15:29:10",
        "tcname": "tc_00187",
      }
    }
  }
}
```

10.1.4 Setting query filters

It is possible to filter the data by specifying query conditions in the URL query parameters. The easiest way to learn how such query strings could look like is to use Search parameters forms in the nliteme web application and then inspect the generated URL query string.

For example, below is the example of how to filter the test results from TSuite_1 test suite executed on test line Line_3 using the build 40.00_2024-10-23-12-13:

Created Between	And	Build
Increment	<input type="text"/>	<input type="text" value="40.00_2024-10-23-12-13"/>
TestSuite	<input type="text" value="TSuite_1"/>	<input type="text"/>
TestLine	<input type="text" value="Line_3"/>	<input type="text"/>
UE Type	<input type="text"/>	<input type="text"/>
Feature	<input type="text"/>	<input type="text"/>

Submit Clear

Test Results (13)

	TestCase	Increment	Build	TestSuite	Verdict	TestLine	Feature	Feature Coverage %	UE Type	Defect	Created On
<input type="checkbox"/>	tc_0099	40.00	40.00_2024-10-23-12-13	TSuite_1	✓	Line_3	1004 Feature_4	5	UE0	N/A	2024-10-23 17:20:15
<input type="checkbox"/>	tc_0095	40.00	40.00_2024-10-23-12-13	TSuite_1	✗	Line_3	1004 Feature_4	5	UE2	1900	2024-10-23 17:25:10
<input type="checkbox"/>	tc_0091	40.00	40.00_2024-10-23-12-13	TSuite_1	✓	Line_3	1004 Feature_4	5	UE1	N/A	2024-10-23 17:27:57
<input type="checkbox"/>	tc_0087	40.00	40.00_2024-10-23-12-13	TSuite_1	✗	Line_3	1004 Feature_4	5	UE0	1900	2024-10-23 17:32:46

The corresponding URL with the json response would look like:

http://localhost/nliteme/index.php?action=com.nliteme.TestResults&usejson&build=40.00_2024-10-23-12-13&tsid%5B%5D=6&tlid%5B%5D=2

As can be seen the filter parameters for test suite and test line are provided in the form of the corresponding indices, which makes the queries more efficient. Additionally, it allows to specify multiple values for the same index type e.g. to filter test results from multiple test lines. For such multiple values selection the [] has to put next to the index name.

[localhost/nliteme/index.php?action=com.nliteme.TestResults&usejson&build=40.00_2024-10-23-12-13&tsid\[\]=%099&tlid\[\]=%2](http://localhost/nliteme/index.php?action=com.nliteme.TestResults&usejson&build=40.00_2024-10-23-12-13&tsid[]=%099&tlid[]=%2)

```

{
  "content": {
    "detailActions": { ... }, // 7 items
    "fieldActions": { ... }, // 1 item
    "checkboxid": "id",
    "label": "TPL_TESTRESULTS",
    "header": { ... }, // 12 items
    "quicklinks": { ... }, // 2 items
    "submitAction": "com.nliteme.TestResults",
    "body": {
      "0": {
        "id": "9192",
        "createdate": "2024-10-23 17:20:15",
        "incid": "4",
        "buildd": "31",
        "tsid": "6",
        "tcid": "202",
        "tlid": "2",
        "tcverdict": "0",
        "extracolumn_0": "0",
        "extracolumn_1": "0",
        "extracolumn_2": "0",
        "extracolumn_3": "0.00",
        "duration": "65",
        "filepath": "ftp://ftp.logs.server/tc_0099_20241023_172015.7z",
        "increment": "40.00",
        "build": "40.00_2024-10-23-12-13",
        "tsname": "TSuite_1",
        "tcname": "tc_0099",
        "fid": "13",
        "coverage": "5",
        "tlname": "Line_3",
        "fname": "1004_Feature_4",
        "hlink": "http://localhost/feature/testfeature?id=1004",
        "history": "http://localhost/nliteme/index.php?tcid=202&incid=4&tsid=6&tlid=2&extracolumn_0=0&action=com.nliteme.MainTestResults&sort=createdate&order=DESC"
      },
      "1": { ... }, // 24 items
      "2": { ... }, // 24 items
      "3": { ... }, // 24 items
      "4": { ... }, // 24 items
      "5": { ... }, // 24 items
      "6": { ... }, // 24 items
      "7": { ... }, // 24 items
      "8": { ... }, // 24 items
      "9": { ... }, // 24 items
    }
  }
}

```

In some case it might be more appropriate to use the named based searches instead of index based, because it allows to use substrings and some wildcards. For example, to filter the test results for the test cases which contain the `099` substring anywhere in the name, the query would look like:

http://localhost/nliteme/index.php?action=com.nliteme.TestResults&usejson&build=40.00_2024-10-23-12-13&tcname=%099

Note the `%` wildcard in front of the substring, which makes the pattern to correspond to the `*reg_wus*` regex. Without it, the pattern would correspond to the `^099*` regex



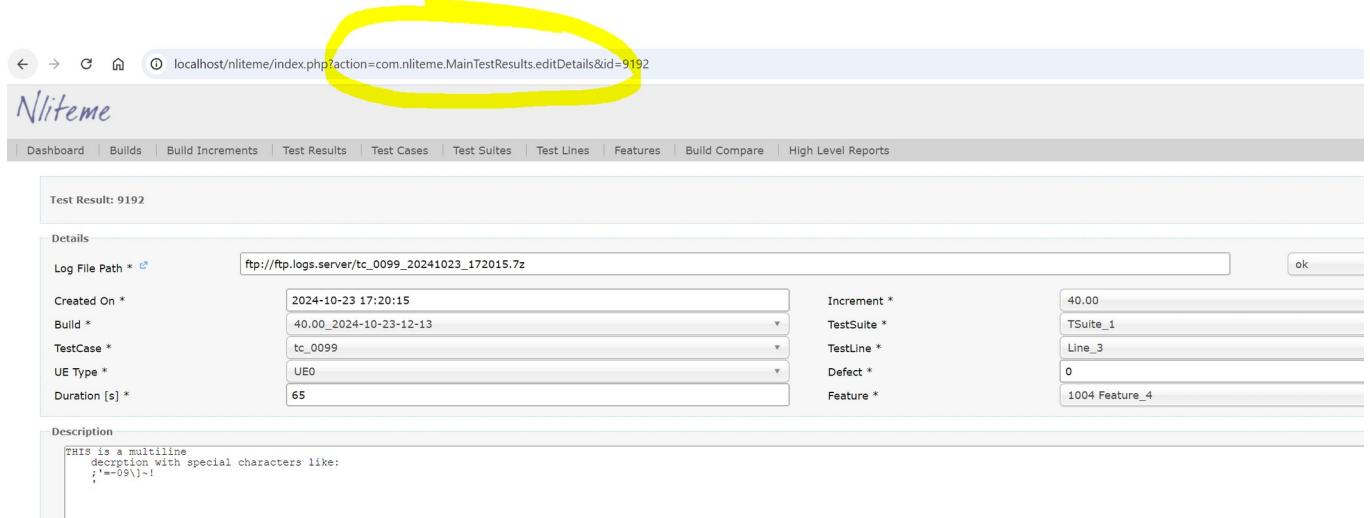
```

{
  "content": {
    "detailActions": [...], // 7 items
    "fieldActions": [...], // 1 item
    "checkboxid": "id",
    "label": "TPL_TESTRESULTS",
    "header": [...], // 12 items
    "quicklinks": [...], // 2 items
    "submitAction": "com.nliteme.TestResults",
    "body": {},
    "numberofrecords": "0",
    "footer": {}
  },
  "messages": {},
  "link": {
    "script": "/nliteme/index.php",
    "server": "http://localhost",
    "query": "action=com.nliteme.TestResults&usejson=&build=40.00_2024-10-23-12-13&tcname=%099",
    "query_array": {
      "action": "com.nliteme.TestResults",
      "usejson": "",
      "build": "40.00_2024-10-23-12-13",
      "tcname": "\t9"
    },
    "url": "http://localhost/nliteme/index.php?action=com.nliteme.TestResults&usejson&build=40.00_2024-10-23-12-13&tcname=%099"
  }
}

```

10.2 Creating queries based on details views

The information available in the detail views can also be retrieved as json response. Similarly, to list views, the easiest way to find out the HTTP URL is to open the web application details views and strip the *Main* prefix from the *action* parameter e.g.



Test Result: 9192

Details

Log File Path *	<input type="text" value="ftp://ftp.logs.server/tc_0099_20241023_172015.7z"/>	ok	
Created On *	<input type="text" value="2024-10-23 17:20:15"/>	Increment *	<input type="text" value="40.00"/>
Build *	<input type="text" value="40.00_2024-10-23-12-13"/>	TestSuite *	<input type="text" value="TSuite_1"/>
TestCase *	<input type="text" value="tc_0099"/>	TestLine *	<input type="text" value="Line_3"/>
UE Type *	<input type="text" value="UE0"/>	Defect *	<input type="text" value="0"/>
Duration [s] *	<input type="text" value="65"/>	Feature *	<input type="text" value="1004 Feature_4"/>

Description

```
THIS is a multiline
description with special characters like:
\';\"=-09\`~!
```

would be converted to:

```
http://localhost/nliteme/index.php?
action=com.nliteme.TestResults.editDetails&id=9192&usejson
```



10.3 Creating queries based on High Level Report views

The principles of URL query parameters, filtering etc. are the same as in the case of list views. Since the high level report use lazy loading, the user has to set the *page* parameter to load next set of data.