Taking advantage of JUnit XML reports

- About JUnit
- JUnit Basic Concepts
- Importing JUnit XML reports
    - Entities
    - Status
- References

About JUnit

blocked URL

JUnit is a testing framework for Java, mostly focused on unit testing.

It is also used for writing integration and acceptance tests, making use of other libraries such as Selenium.

JUnit was massively used by the Java community and thus, its XML test result reports have become a de facto standard for test result reporting.

JUnit XML reports may be created by many different testing frameworks for Java, JavaScript, Ruby, Python, or any other language.

JUnit Basic Concepts

In JUnit, you have Tests and (Test) Suites. A Suite is a way of aggregating a group of tests together, along with their results. This applies not just to the original Java's JUnit but also for other implementations that generate the JUnit XML report.

In Java, Tests are created within a Test Case class which will contain the Tests, implemented as class methods (and properly annotated).

The Test Case classes may be grouped in Test Suites.

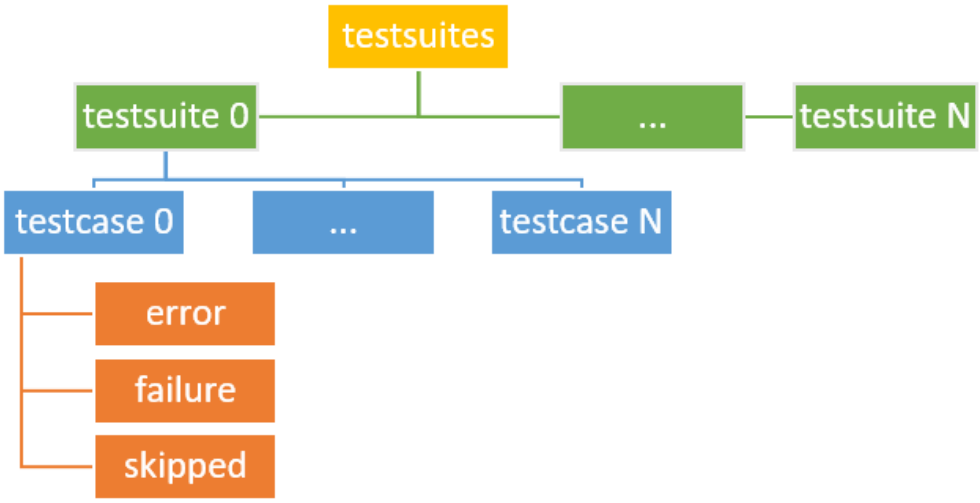
JUnit provides way more concepts (Test Runners, Test Fixtures, Categories, etc.) although they are not relevant in this context.

Importing JUnit XML reports

Below is a simplified example of a JUnit XML report containing a Test Suite with one Test Case.

```
    <?xml version="1.0" ?>
<testsuites>
    <testsuite errors="0" failures="0" id="0" name="my test suite" tests="1">
        <testcase classname="some.class.name" name="Test1" time="123.345000"/>
    </testsuite>
</testsuites>
```

The simplified tags hierarchy of these reports can be represented in the following diagram:

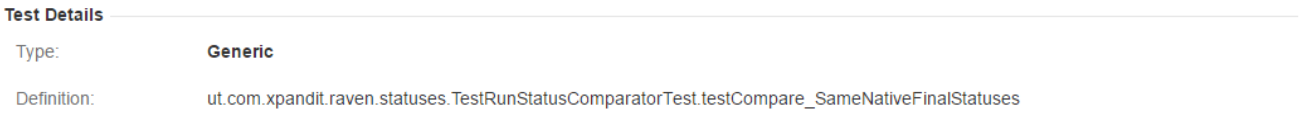


Entities

JUnit's Test Cases are identified by the pair of attributes "classname" and "name" attributes.

Test Cases are imported to Xray's ***Generic* Test issues**, and the "classname" and "name" attributes are concatenated and mapped to the **Generic Test Definition** field of the Generic Test.

If a Test already exists with the same **Generic Test Definition**, then it is not created again.



Test Cases are imported to a new (or user-specified) Test Execution in the context of some project, along with their respective execution results.

JUnit's Test Suites are not mapped to any special entity. However, the execution details screen will show the Test Suite related to a specific test result.

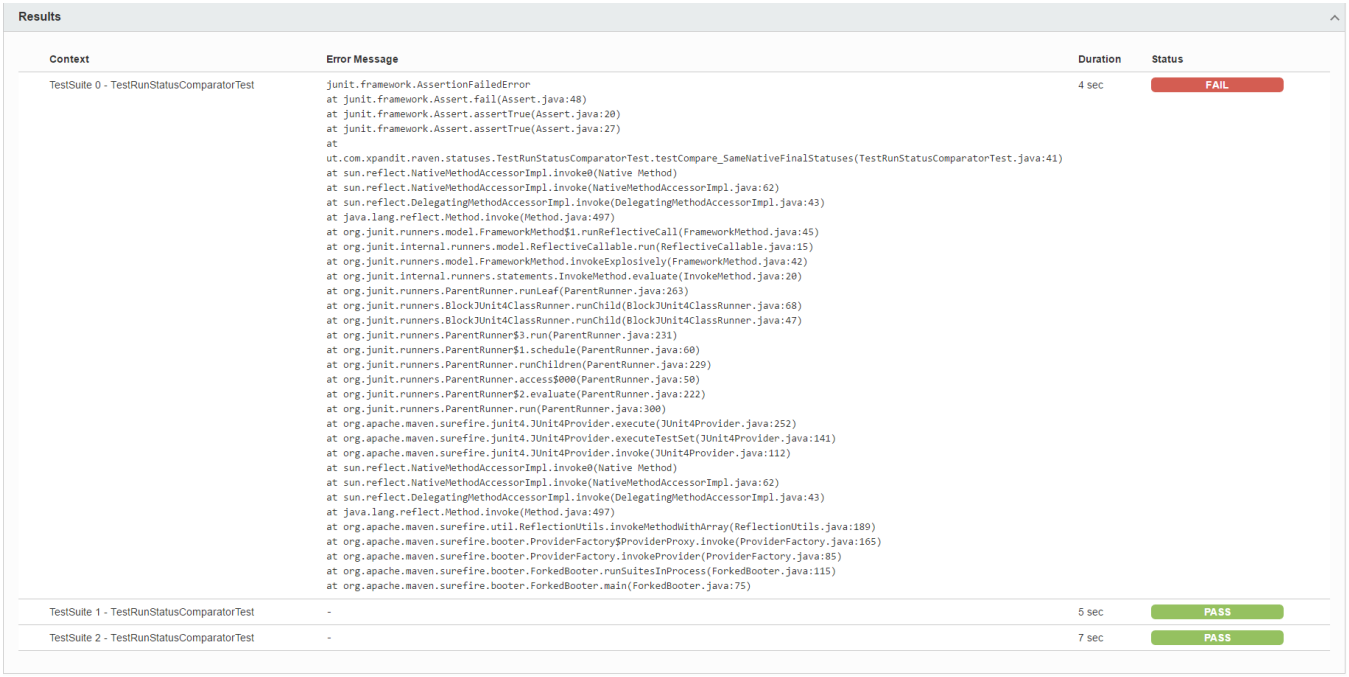
Status

The status of the Test Run will be set based on the Test Case result:

Test Cases	Test status
with failures	FAIL
with errors	FAIL
skipped	TODO
without failures, errors, and weren't skipped	PASS

Note: Test Cases with the status FAIL may have an error/failure message which can be seen in the Test Run screen, under the Results section.

If the same Test Case has been executed on multiple Test Suites, then the result for each Test Suite will be shown.



When a Test Case is executed in multiple Test Suites, the overall status of the Test Run will be calculated as a joint value.

Condition	Overall status of the Test Run
If all the mapped results of the Test Case was PASS	**PASS**
If any of the mapped results of the Test Case was FAIL	**FAIL**
Other cases	**TODO**

References

- https://github.com/junit-team/junit4/wiki
- http://junit.org/junit4/
- https://www.tutorialspoint.com/junit/junit_basic_usage.htm
- https://www.relishapp.com/cucumber/cucumber/docs/formatters/junit-output-formatter