
Squash TF Java Junit Runner Documentation

squashtest

Apr 01, 2020

Contents

1	Overview	1
2	Contents	3
2.1	Usage	3

CHAPTER 1

Overview

The **Squash Test Factory (Squash TF)** Java Junit Runner aims to provide a seamless integration to our ecosystem when automated test are already (will be) implemented using Java as a language and Junit as the underlying test framework.

As a **Squash TF** runner its main goal is twofold :

- List in JavaScript Object Notation (Json) format the available implemented tests
- Run a selection (that can include all available tests) and report the execution. In the case where the execution order originates from **Squash Test Management (Squash TM)**, test status and report are sent back to **Squash TM**.

The actual implementation of the Runner is based on the Maven technology. More precisely each major feature (“List” and “Run”) is provided via an implementation of a Maven Mojo.

2.1 Usage

The runner is transparent and should be able to run any Maven Junit 4 or 5 project **without the necessity to modify it**. Indeed the Mojos are implemented in such a way that they are fully autonomous and do not need any project specific configuration to run.

2.1.1 List implemented Junit tests

This Mojo enables one to list as a Json file the available implemented tests. In order to do so one only needs to run the following command :

```
mvn clean compile test-compile org.squashtest.ta.galaxia:squash-tf-junit-runner-maven-  
→plugin:1.0.0-RELEASE:list
```

The command is structured as follows

- `mvn` : launch Maven
- `clean` : (Optional) One of Maven default goals that enables one to clean everything that has been previously build by Maven.
- `compile` : One of Maven default goals that enables one to compile code that is stored in `src/main`.
- `test-compile` : One of Maven default goals that enables one to compile code that is stored in `src/test`.
- `org.squashtest.ta.galaxia:squash-tf-junit-runner-maven-plugin:1.0.0-RELEASE:list` : the actual listing Mojo provided by **Squash TF** Java Junit Runner. It lists all Junit tests that can be discovered (in the Junit sense) in the code compiled during the “compile” and “test-compile” phases.

The result should be a Json file named `testTree.json` located at `target/squashTA/test-tree`. It is a simple JavaScript table listing availables test grouped by support classes. The name of the “ecosystems” follows the following convention “name-of-bundle:qualified.class.name”, where “name-of-bundle” is the name of the Maven bundle where the test was found and “qualified.class.Name” is the qualified class name where the test is implemented. Currently supported

bundle names are : * `maven.main.bundle` : Name given to the Maven bundle that can be found at “src/main” * `maven.test.bundle` : Name given to the Maven bundle that can be found at “src/test” Finally the name of the test is the Junit “Display name”.

2.1.2 Run Junit tests

This Mojo enables one to run a selection of, or all possible, Junit tests and report their execution. In order to do so one only needs to run the following command :

```
mvn clean compile test-compile org.squashtest.ta.galaxia:squash-tf-junit-runner-maven-  
↪plugin:1.0.0-RELEASE:run
```

- `mvn` : launch Maven
- `clean` : (Optional) One of Maven default goals that enables one to clean everything that has been previously build by Maven.
- `compile` : One of Maven default goals that enables one to compile code that is stored in `src/main`.
- `test-compile` : One of Maven default goals that enables one to compile code that is stored in `src/test`.
- `org.squashtest.ta.galaxia:squash-tf-junit-runner-maven-plugin:1.0.0-RELEASE:run` : the actual running Mojo provided by **Squash TF Java Junit Runner**. It runs the Junit tests that can be discovered (in the Junit sense) in the code compiled during the “compile” and “test-compile” phases.

By default the whole collection of tests available in the project will be executed. A summary of the execution is reported and available at `target/squashTA/html-reports/squash-ta-report.html` . A more detailed version of the report providing context in the case of technical error is also produced and available at `target/squashTA/html-details/squash-ta-report.html`. Finally a surfire report is also produced and available at `target/squashTA/surefire-reports/` .

if one wants to noly run a subset of possible test one can provide a list of tests via the Maven property “`tf.test.suite`”. Two mechanism are possible:

- Mimic TM-TF link and provide a list of selected test via a Json file. In this scenario the `tf.test.suite` parameter should be given the value “`{file:testsuite.json}`” and the `testsutie.json` should be put right to the project pom.
- Provide a CSV like line, where tests names are listed serated by semicolons

In both cases test convention should follow the one used by the listing Mojo and described in the section above.

2.1.3 Starting a new project

In case you are starting a new Maven Java Junit project from scratch, a Maven archetype is provided to help you generate a correctly structured project with recommended dependencies. The archetype group Id is `org.squashtest.ta.galaxia`, the archetype artifact Id is `squash-tf-junit-runner-project-achetype` and the current version is `1.0.0-RELEASE`. Hence, to use it simply run the following command :

```
mvn archetype:generate -DarchetypeGroupId=org.squashtest.ta.galaxia -  
↪DarchetypeArtifactId=squash-tf-junit-runner-project-archetype -DarchetypeVersion=1.  
↪0.0-RELEASE
```