Pangolin TestRail Connector

Agiletestware Pangolin TestRail Connector plugin integrates Jenkins with TestRail by automatically publishing test results from Jenkins to TestRail using Pangolin server (http://www.agiletestware.com/pangolin/).

The plugin allows users to integrate TestRail into their CI workflow without writing a single line of code.

Plugin Information
View Agiletestware Pangolin Connector for TestRail on the plugin site for more information.

Older versions of this plugin may not be safe to use. Please review the following warnings before using an older version:

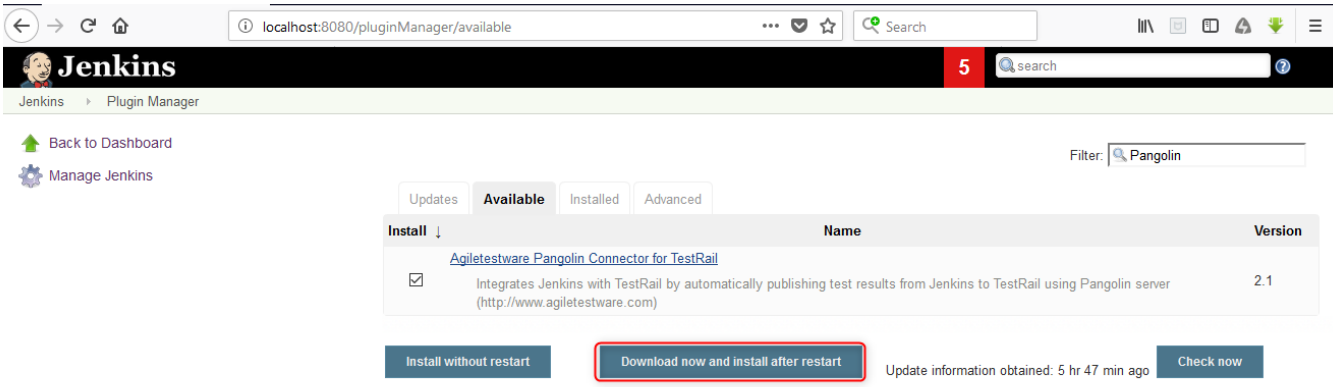
- CSRF vulnerability and missing permission checks allowed overriding plugin configuration

Prerequisites

- Pangolin Server must be installed and accessible from Jenkins host

Jenkins plugin Installation

Installation of Jenkins plugin can be easily done from Jenkins Plugin Manager:



localhost:8080/pluginManager/available

Jenkins › Plugin Manager

Back to Dashboard

Manage Jenkins

Filter: Pangolin

Updates | **Available** | Installed | Advanced

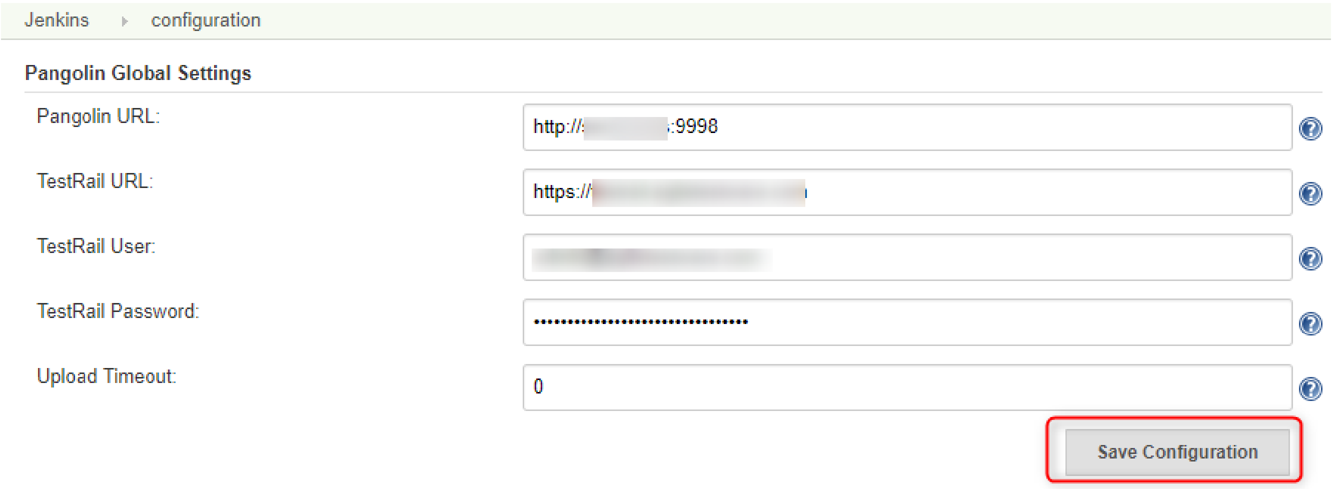
Install ↓	Name	Version
☑	Agiletestware Pangolin Connector for TestRail Integrates Jenkins with TestRail by automatically publishing test results from Jenkins to TestRail using Pangolin server (http://www.agiletestware.com)	2.1

Update information obtained: 5 hr 47 min ago

Please restart Jenkins Server after the installation of plugin

Pangolin Global Configuration

Pangolin global configuration is accessible on "Mange Jenkins" - "Configure System" screen:



Jenkins ▸ configuration

Pangolin Global Settings

Pangolin URL: http://[redacted]:9998

TestRail URL: https://[redacted]

TestRail User: [redacted]

TestRail Password: ••••••••••••••••••••••••••••••••

Upload Timeout: 0

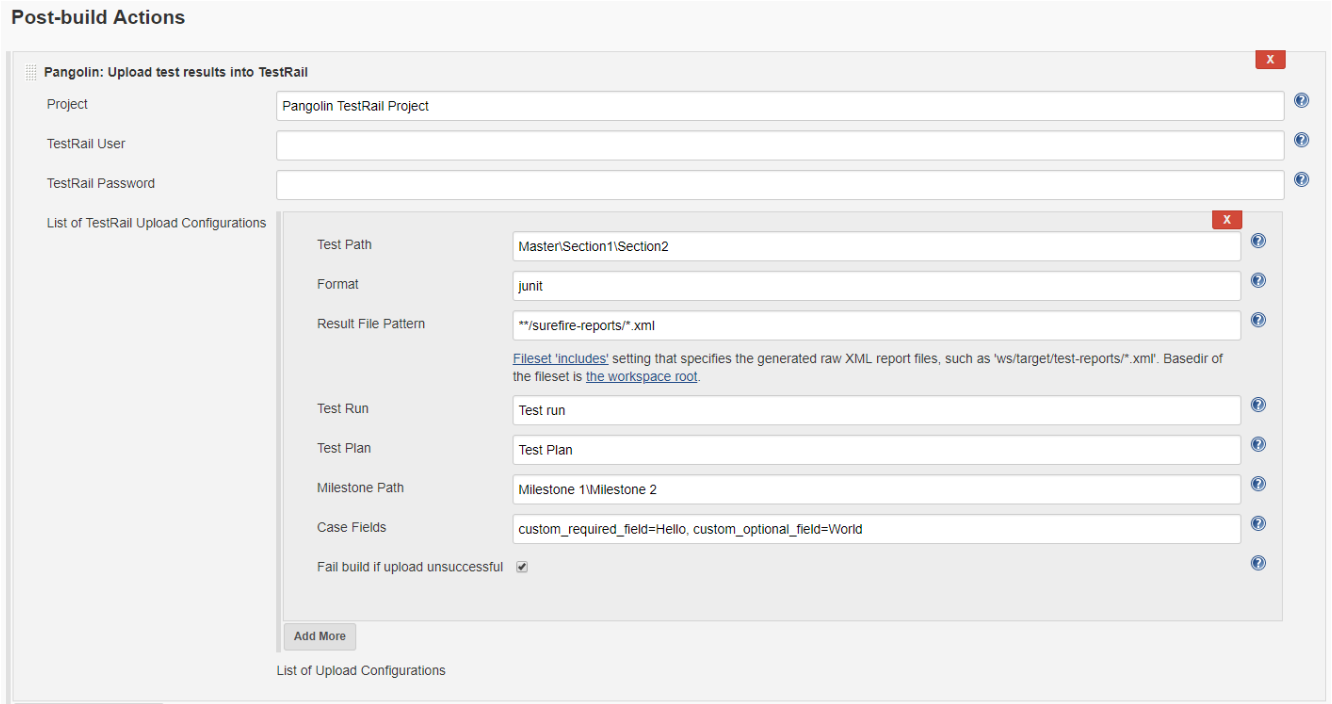
Field Name	Description
Pangolin URL	URL for Pangolin server. Example: http://some-server:8888
TestRail URL	URL for your TestRail instance. Example: https://testrail:8080
TestRail User	Name of user in TestRail
TestRail Password	Password for TestRail user
Upload Timeout	The number of minutes to wait for the Pangolin server to process the request. 0 means wait indefinitely

When all fields are set, please click on "Save Configuration" button - Pangolin will check availability of entered URLs and encrypt your password.

Uploading test framework reports into TestRail

"Pangolin: Upload test results into TestRail" post-build action configuration

To upload test results into TestRail, a new "Pangolin: Upload test results into TestRail" post-build action has to be added:



Post-build Actions

Pangolin: Upload test results into TestRail

Project: Pangolin TestRail Project

TestRail User:

TestRail Password:

List of TestRail Upload Configurations

Test Path: Master\Section1\Section2

Format: junit

Result File Pattern: **/surefire-reports/*.xml

Fileset 'includes' setting that specifies the generated raw XML report files, such as 'ws/target/test-reports/*.xml'. Basedir of the fileset is the workspace root.

Test Run: Test run

Test Plan: Test Plan

Milestone Path: Milestone 1\Milestone 2

Case Fields: custom_required_field=Hello, custom_optional_field=World

Fail build if upload unsuccessful ☑

List of Upload Configurations

Field Name (pipeline property)	Required	Description
Project (testRailProject)	Yes	The name of project in TestRail to which results should be exported
TestRail User (testRailUserName)	No	If specified, overrides TestRail user name defined in Global Configuration
TestRail Password (testRailPassword)	No	If specified, overrides TestRail user password defined in Global Configuration
Test Path (testPath)	Yes	Path to where Pangolin should test definitions, must contain suite name in the beginning (for single-suite project, suite name is always 'Master'), e.g. Master\Section1\Section2
Format (format)	Yes	Format of test results generated during the build. Available values: junit, nunit, testng, trx, cucumber, robot
Results File Pattern (resultPattern)	Yes	Provide path to the report file(s) generated during the build. This is a comma separated list of test result directories. You can also use Ant style patterns such as ** /surefire-reports/*.xml
Test Run (testRun)	No	Name of test run in TestRail to which test results will be added
Test Plan (testPlan)	No	Name of test plan in TestRail to which test results will be added
Milestone Path (milestonePath)	No	Path to a milestone in TestRail to which test run/plan will be added. E.g. Milestone1\Milestone2
Case Fields (customProperties)	No	Values for custom case fields in TestRail. The format is [TestRail field system name]=[value 1], [TestRail field system name]=[value 2]... E.g.: custom_user_field1=foo, custom_user_field2=bar
Fail build if upload unsuccessful (failIfUploadFailed)	N/A	If checked, the build will be marked as failed if for any reason the plugin was not able to upload the results. This could be due to Pangolin server issues, TestRail server issues, network issues, etc.

Several configuration can be specified

To add more than one upload configuration, just click on "Add More" button.

"Pangolin: Upload test results into TestRail" pipeline configuration

Here is an example of pipeline configuration for exporting test results to TestRail:

Pipeline configuration

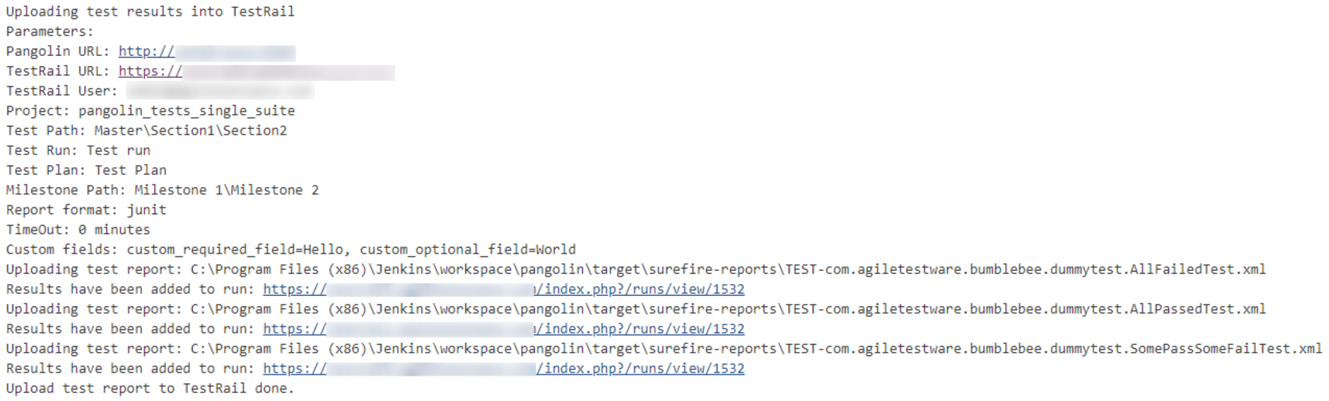
```
pangolinTestRail([
     testRailProject: 'project',
     testRailUserName: 'user',
     testRailPassword: 'encrypted password',
        configs:[[
         testPath: 'Master\\Section1\\Section2',
         format: 'JUNIT',
         resultPattern:'**/surefire-reports/*.xml',
         testRun:'test run',
         testPlan:'test plan',
         milestonePath:'milestonePath',
         customProperties:'custom_required_field=foo',
         failIfUploadFailed: true
         ]],
       ])
```

TestRail password must be encrypted

When pipeline script is used, value for testRailPassword must be encrypted. Password can be encrypted by navigating to Pangolin Server GUI page (http://pangolinUrl) and clicking to Password Encryption link.

Execution

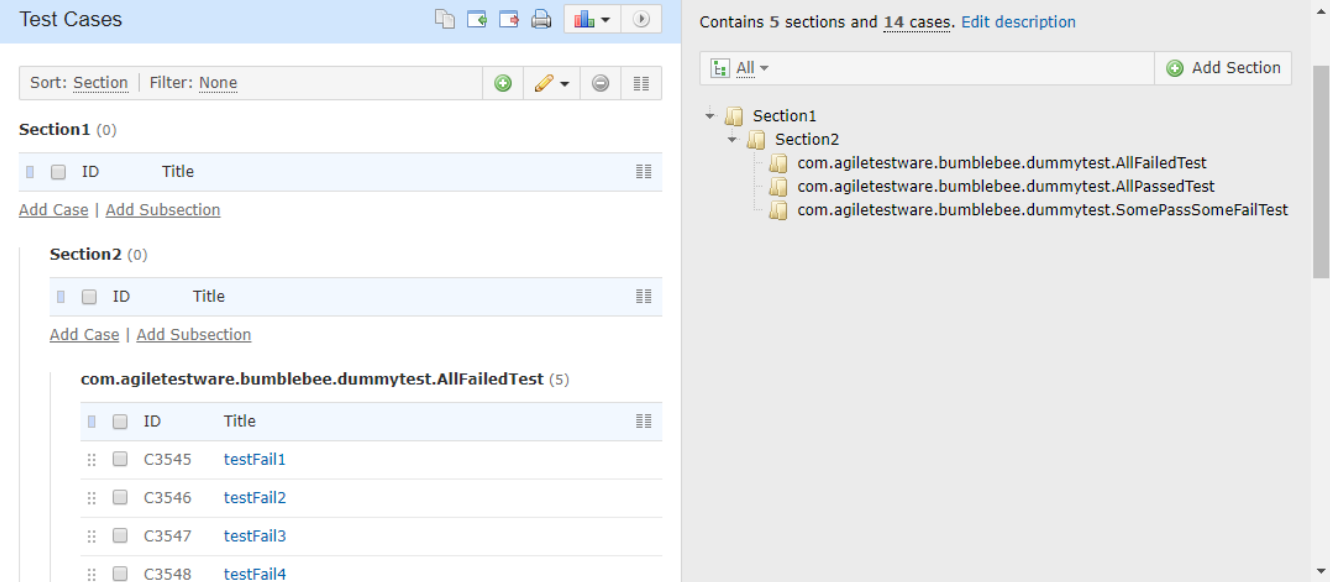
Once the job is configured, simply run the job. Here is Jenkins job output example:



```
Uploading test results into TestRail
Parameters:
Pangolin URL: http://[redacted]
TestRail URL: https://[redacted]
TestRail User: [redacted]
Project: pangolin_tests_single_suite
Test Path: Master\Section1\Section2
Test Run: Test run
Test Plan: Test Plan
Milestone Path: Milestone 1\Milestone 2
Report format: junit
TimeOut: 0 minutes
Custom fields: custom_required_field=Hello, custom_optional_field=World
Uploading test report: C:\Program Files (x86)\Jenkins\workspace\pangolin\target\surefire-reports\TEST-com.agiletestware.bumblebee.dummytest.AllFailedTest.xml
Results have been added to run: https://[redacted]/index.php?/runs/view/1532
Uploading test report: C:\Program Files (x86)\Jenkins\workspace\pangolin\target\surefire-reports\TEST-com.agiletestware.bumblebee.dummytest.AllPassedTest.xml
Results have been added to run: https://[redacted]/index.php?/runs/view/1532
Uploading test report: C:\Program Files (x86)\Jenkins\workspace\pangolin\target\surefire-reports\TEST-com.agiletestware.bumblebee.dummytest.SomePassSomeFailTest.xml
Results have been added to run: https://[redacted]/index.php?/runs/view/1532
Upload test report to TestRail done.
```

Results in TestRail:

Test Cases:



Test Cases

Sort: Section | Filter: None

Section1 (0)

ID	Title

Add Case | Add Subsection

Section2 (0)

ID	Title

Add Case | Add Subsection

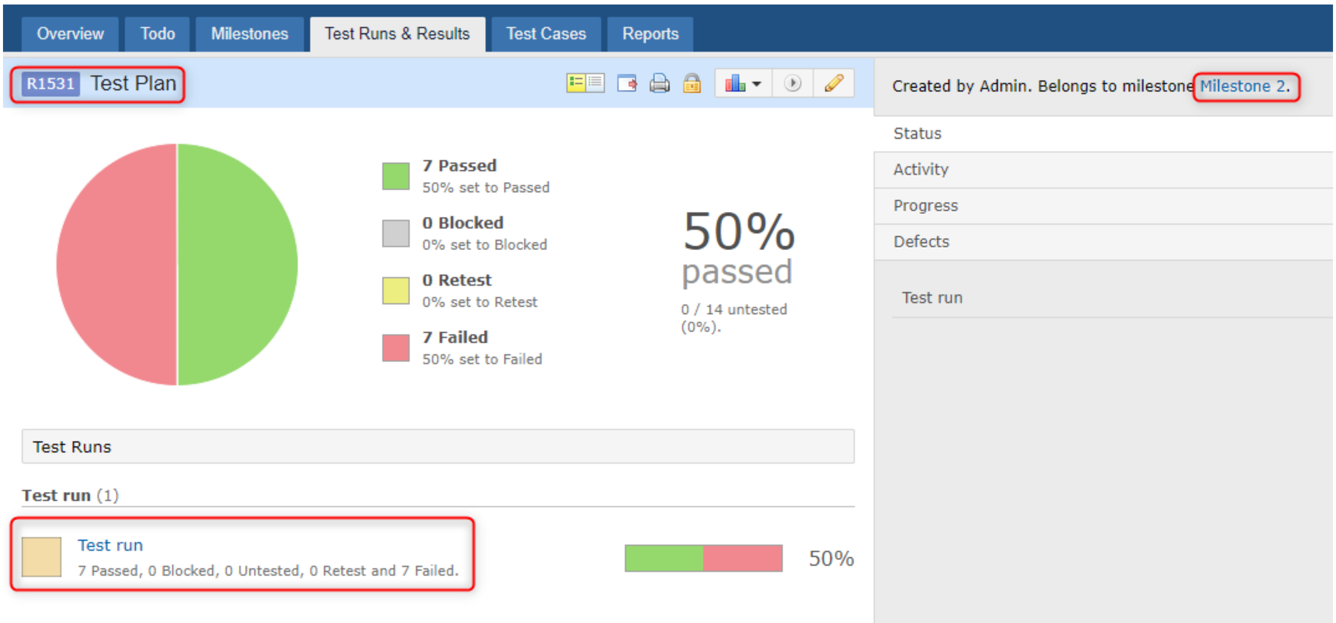
com.agiletestware.bumblebee.dummytest.AllFailedTest (5)

ID	Title
C3545	testFail1
C3546	testFail2
C3547	testFail3
C3548	testFail4

Contains 5 sections and 14 cases. Edit description

- Section1
  - Section2
    - com.agiletestware.bumblebee.dummytest.AllFailedTest
    - com.agiletestware.bumblebee.dummytest.AllPassedTest
    - com.agiletestware.bumblebee.dummytest.SomePassSomeFailTest

Test Plan and Test Run:



R1531 Test Plan

Created by Admin. Belongs to milestone Milestone 2.

🟩 **7 Passed** 50% set to Passed

▢ **0 Blocked** 0% set to Blocked

🟨 **0 Retest** 0% set to Retest

🟥 **7 Failed** 50% set to Failed

50% passed

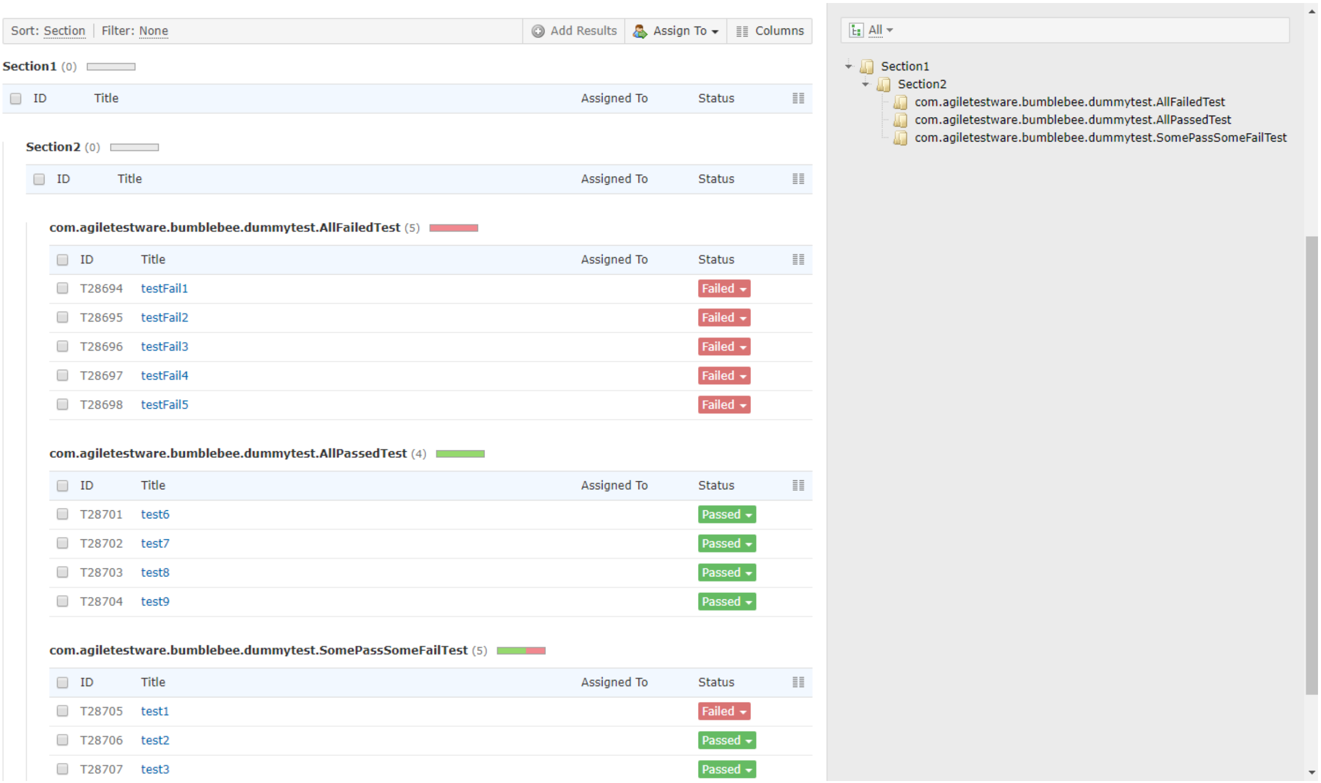
0 / 14 untested (0%).

Test Runs

Test run (1)

Test run
7 Passed, 0 Blocked, 0 Untested, 0 Retest and 7 Failed.

50%



Sort: Section | Filter: None

Section1 (0)

□	ID	Title	Assigned To	Status

Section2 (0)

□	ID	Title	Assigned To	Status

com.agiletestware.bumblebee.dummytest.AllFailedTest (5)

□	ID	Title	Assigned To	Status
□	T28694	testFail1		Failed
□	T28695	testFail2		Failed
□	T28696	testFail3		Failed
□	T28697	testFail4		Failed
□	T28698	testFail5		Failed

com.agiletestware.bumblebee.dummytest.AllPassedTest (4)

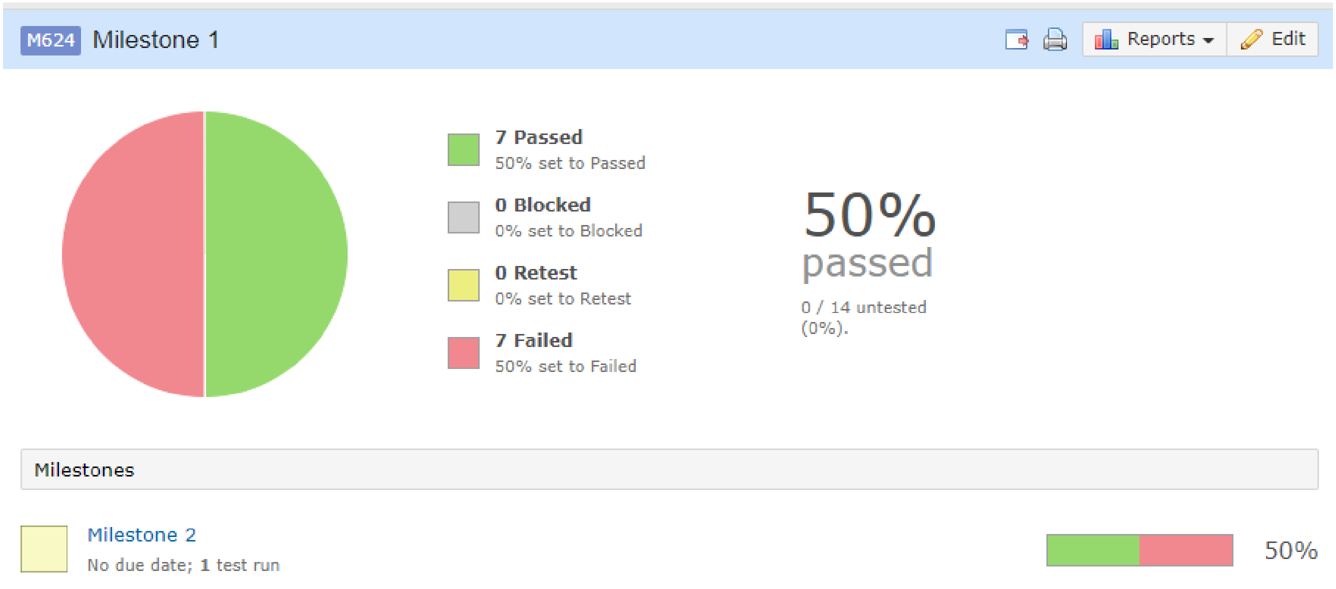
□	ID	Title	Assigned To	Status
□	T28701	test6		Passed
□	T28702	test7		Passed
□	T28703	test8		Passed
□	T28704	test9		Passed

com.agiletestware.bumblebee.dummytest.SomePassSomeFailTest (5)

□	ID	Title	Assigned To	Status
□	T28705	test1		Failed
□	T28706	test2		Passed
□	T28707	test3		Passed

- Section1
  - Section2
    - com.agiletestware.bumblebee.dummytest.AllFailedTest
    - com.agiletestware.bumblebee.dummytest.AllPassedTest
    - com.agiletestware.bumblebee.dummytest.SomePassSomeFailTest

Created Milestones hierarchy:



M624 Milestone 1

🟩 **7 Passed**
50% set to Passed

▨ **0 Blocked**
0% set to Blocked

🟨 **0 Retest**
0% set to Retest

🟥 **7 Failed**
50% set to Failed

50% passed

0 / 14 untested (0%).

Milestones

Milestone 2
No due date; **1** test run

50%

Changelog

Version 2.3 (Released Nov, 2018)

- Add support for Cucumber JSON reports
- Add support for Robot Framework xUnit reports
- Upload NUnit report attachments
- Publish TestRail Run link to build page

Version 2.2 (Released July, 2018)

- Security fix for CSRF vulnerability and missing permission checks (SECURITY-995)

Version 2.1 (Released May, 2018)

- The first public release