

# Testopia Plugin

## Plugin Information

View Testopia [on the plugin site](#) for more information.



This plugin is up for adoption. Want to help improve this plugin? [Click here to learn more!](#)

This plug-in integrates **Jenkins** with Testopia and generates reports on automated test execution. With this plug-in you can manage your tests in Testopia, schedule and control in **Jenkins**, and execute using your favorite test execution tool (TestPartner, Selenium, TestNG, Perl modules, prove, PHPUnit, among others).



When filing an issue, remember to include as much data as possible, but avoid including sensitive data, as attachments can take a while to be deleted from JIRA, as the process is manual and quite troublesome 🙄

## Description

The Testopia plug-in integrates **Jenkins** with [Testopia](#). Testopia plug-in uses [testopia-java-driver](#) to access Testopia **XML-RPC** API. With the information that you provide in the build step configuration the plug-in **retrieves automated tests from Testopia**. With the plug-in, you are able to execute **build steps** that call testing tools. It reads **TestNG**, **JUnit** and **TAP** test report **formats**, used to update Testopia test cases' executions.

## Testopia configuration section

Configure System [jen] x  
localhost:8080/configure

**Subversion**  
Subversion Workspace Version: 1.4  
Exclusion revprop name:   
 Validate repository URLs up to the first variable name

**Shell**  
Shell executable:

**Testopia**  
Testopia Installation  
Name:   
URL:   
User name:   
Password:   
Advanced...  
Delete  
Add  
List Of Testopia Installations

**Jenkins URL**  
Jenkins URL:

**E-mail Notification**  
SMTP server:   
Default user e-mail suffix:   
System Admin E-mail Address:

## Test Execution section

testopia-junit Config

localhost:8080/job/testopia-junit/configure

**Invoke Testopia**

**Testopia Configuration**

Testopia Version:

Test Run ID:

**Test Execution**

Single Test Build Steps:

Iterative Test Build Steps:

- Execute shell**
- Command:
- [See the list of available environment variables](#)
- 
- 
- 

**Result Seeking Strategy**


Test Result Seeking Strategies:

- JUnit method name**
- Include Pattern:
- 
- 
- 

**Post-build Actions**

Aggregate downstream test results

### Result Seeking Strategy section

 You can use environment variables.

## How does the plug-in execute my automated tests?

Testopia Plug-in retrieves automated test cases data from Testopia. This data is then set as environment variables. This way your build steps can make use of this information to execute your automated tests.

For instance, you could run a ant command passing a Test Class as parameter, more or less something like this, `/opt/java/apache-ant-1.8.0/bin/ant -DtestClass=$TESTOPIA_TESTCASE_SCRIPT`. In this case, `TESTOPIA_TESTCASE_SCRIPT` the value of the script automation field in Testopia.

List of environment variables available during Testopia Plug-in execution:

### Already implemented:

#### Test run info

- TESTOPIA\_TESTRUN\_ID
- TESTOPIA\_TESTRUN\_BUILD
- TESTOPIA\_TESTRUN\_ENVIRONMENT
- TESTOPIA\_TESTRUN\_MANAGER
- TESTOPIA\_TESTRUN\_NOTES
- TESTOPIA\_TESTRUN\_PRODUCT\_VERSION
- TESTOPIA\_TESTRUN\_SUMMARY
- TESTOPIA\_TESTRUN\_CASES
- TESTOPIA\_TESTRUN\_PLAN\_ID
- TESTOPIA\_TESTRUN\_PLAN\_TEXT\_VERSION
- TESTOPIA\_TESTRUN\_RUN\_ID
- TESTOPIA\_TESTRUN\_STATUS
- TESTOPIA\_TESTRUN\_TARGET\_COMPLETION
- TESTOPIA\_TESTRUN\_PLAN\_TARGET\_PASS

#### Test case info

- TESTOPIA\_TESTCASE\_ID
- TESTOPIA\_TESTCASE\_RUN\_ID
- TESTOPIA\_TESTCASE\_BUILD\_ID

- TESTOPIA\_TESTCASE\_SCRIPT
- TESTOPIA\_TESTCASE\_ALIAS
- TESTOPIA\_TESTCASE\_ARGUMENTS
- TESTOPIA\_TESTCASE\_REQUIREMENT
- TESTOPIA\_TESTCASE\_SORT\_KEY
- TESTOPIA\_TESTCASE\_SUMMARY
- TESTOPIA\_TESTCASE\_AUTHOR\_ID
- TESTOPIA\_TESTCASE\_CATEGORY\_ID
- TESTOPIA\_TESTCASE\_DEFAULT\_TESTER\_ID
- TESTOPIA\_TESTCASE\_PRIORITY\_ID
- TESTOPIA\_TESTCASE\_STATUS\_ID
- TESTOPIA\_TESTCASE\_AUTOMATED
- TESTOPIA\_TESTCASE\_CREATION\_DATE
- TESTOPIA\_TESTCASE\_ESTIMATED\_TIME
- TESTOPIA\_TESTCASE\_ESTIMATED\_ENV\_ID

### How does the plug-in know if a test passed or failed?

You can choose among different **result seeking strategies** in your job configuration. The plug-in will support three **result formats**, *TestNG*, *JUnit* and *TAP*.

#### Already implemented:

- [TAP](#) (using [tap4j](#))
- JUnit (using Jenkins built-in parser)
- TestNG (using [testng-parser](#))

TAP result seeking strategies

**TAP file name:** The plug-in matches the TAP file name (e.g.: testFtpProtocol.tap) and the key custom field value.

#### Does the plugin record the test results in Testopia?

Yes, the plugin will update the testcase in Testopia and set the status according to the result of the testcase.

## Configuration

1. Download and install the latest version of Testopia (<http://www.mozilla.org/projects/testopia>).
2. Create automated tests in Testopia (Check automated field).
3. Install the Testopia plug-in from the Jenkins Plugin Manager
4. Define a Testopia configuration from the Configure System page.
5. Add a Invoke Testopia build step in the job you want execute automated tests.
6. Configure the required properties.

## Languages Supported

1. English (American)
2. Dutch (the Netherlands)
3. Portuguese (Brazil)

Want to see this plug-in in your language? Send us an e-mail and we will get in touch with the text that needs to be translated.

## Compatibility Matrix




We don't maintain the plug-in for Hudson. Issues happening in Hudson may not be fixed here.

Plugin	1.0	1.1	1.2	1.3
Testopia	2.5+	2.5+	2.5+	2.5+
Jenkins	1.424 +	1.424 +	1.424 +	1.424 +

## Issues

type key summary assignee reporter priority status resolution created updated due

 Can't show details. Ask your admin to whitelist this Jira URL.

[View these issues in Jira](#)

## Resources

Testopia Plug-in used TestLink Plug-in as basis. Below you will find resources for Testopia and TestLink plug-ins, as the concept in both plug-ins is very similar.

1. [Article](#) published in [Testing Experience magazine](#) issue number 12 (Open Source Tools) written by [Bruno P. Kinoshita](#) and Anderson dos Santos. 2010.
2. [Slides](#) used in the lightning talk presented at [Encontro Ágil 2010](#) in [IME-USP](#) (Portuguese Only). 2010.
3. [Article](#) published in [Automated Software Testing Magazine](#), volume 3, issue 1. March, 2011.
4. [Slides](#) used in the presentation done in [Belgium Testing Days](#) 2011.
5. [Lighting talk](#) for [Encontro Ágil 2010](#), at [IME-USP](#).
6. More articles and tips on [Bruno P. Kinoshita's website](#).
7. Presentation at [STPCon](#) Spring March 2012 in New Orleans - USA.
8. [Automated Testing Environment by Bugzilla, Testopia and Jenkins](#) by Chin Pin Chang. July, 2014.

## Documentation

1. Various Testopia documentation assembled on [Storify](#)
2. How to handle self signed certificates on Jenkins Server ([TestLink Plugin SSL support](#))

## Sponsors



## Release Notes

### Release 1.3

1. JUnit and TestNG result seeking strategies

### Release 1.0

1. First version of the plug-in (the development was guided based on the source code of the following plug-ins: [TestLink CCM](#) , [Sonar](#) (how to ref maven installations)).

## Roadmap

1. Keep compatibility with latest versions of Jenkins and Testopia.

## Troubleshooting

- A test case without a tester assigned doesn't execute.