

Integrate Selenium with Jenkins

JUNE 13, 2018 NIDHI GUPTA 1 COMMENT

How to Integrate Selenium with Jenkins

This blog will help you to Integrate Selenium with Jenkins. This will help you to run the selenium scripts using Jenkins automatically or as per your need.

Requirement

1. Download the sample code from github
2. Maven Installation
3. Setup with eclipse

Download the sample code from github

1. Clone the repository from here
2. cd to the location
3. run the mvn test on your machine.

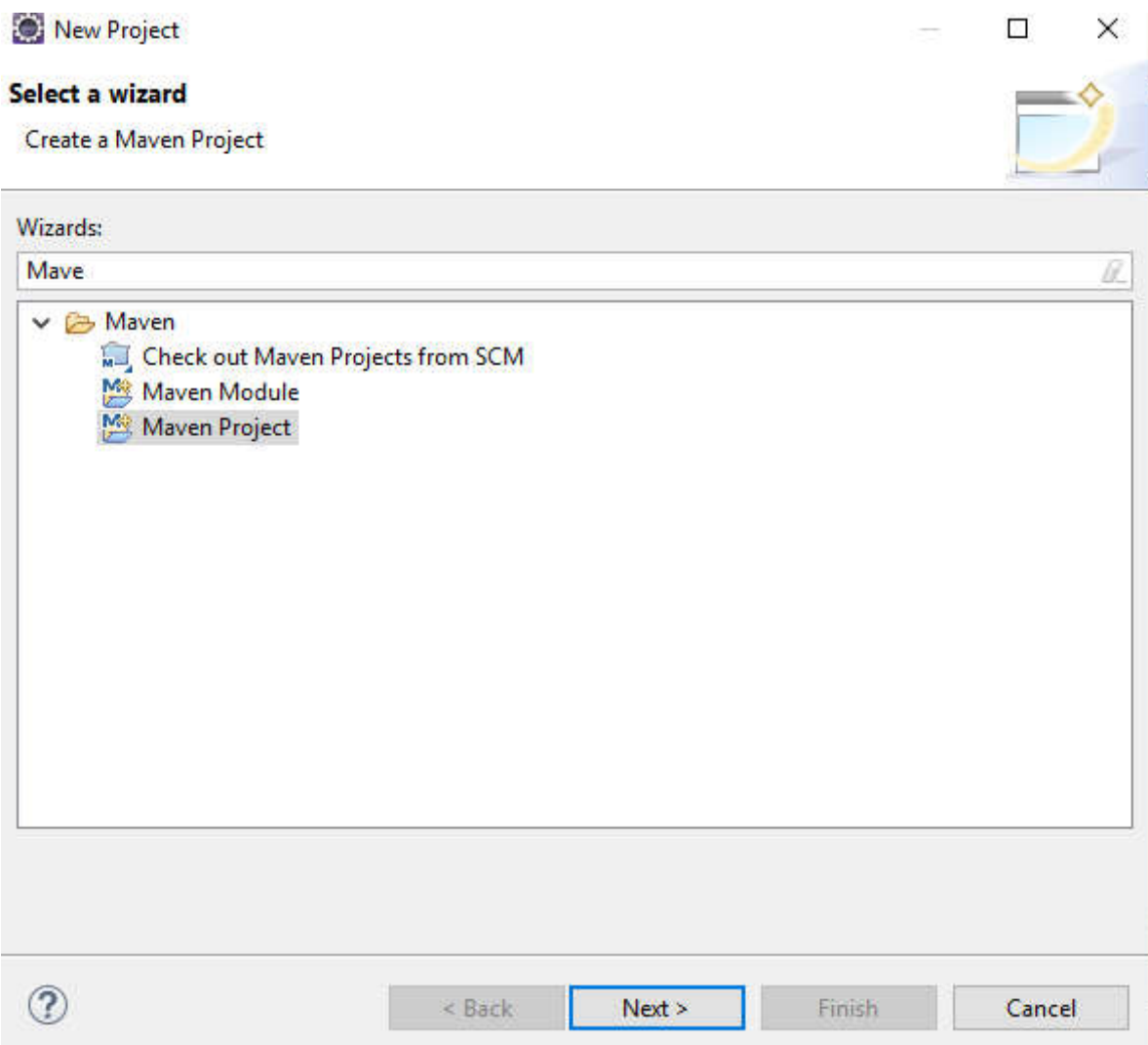
Jenkins Configuration

1. Create a freestyle job, in scm—use the link mentioned above -> in build section -> add clean test -> save -> trigger

Setup with eclipse

See the below steps if you want to know how to configure a sample project in eclipse to run selenium scripts.

1. Download eclipse from here
2. Create a folder on your machine C://Workspace
3. Launch eclipse.exe (Note java8 is required)
4. Create a new Maven Project



6. Click Next and select create a sample project

New Maven Project

New Maven project

Select project name and location

☒ Create a simple project (skip archetype selection)

☒ Use default Workspace location

Location:

☐ Add project(s) to working set

Working set:

▶ Advanced

New Maven Project

New Maven project

Configure project

Artifact

Group Id:

Artifact Id:

Version:

Packaging:

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version:

▶ Advanced

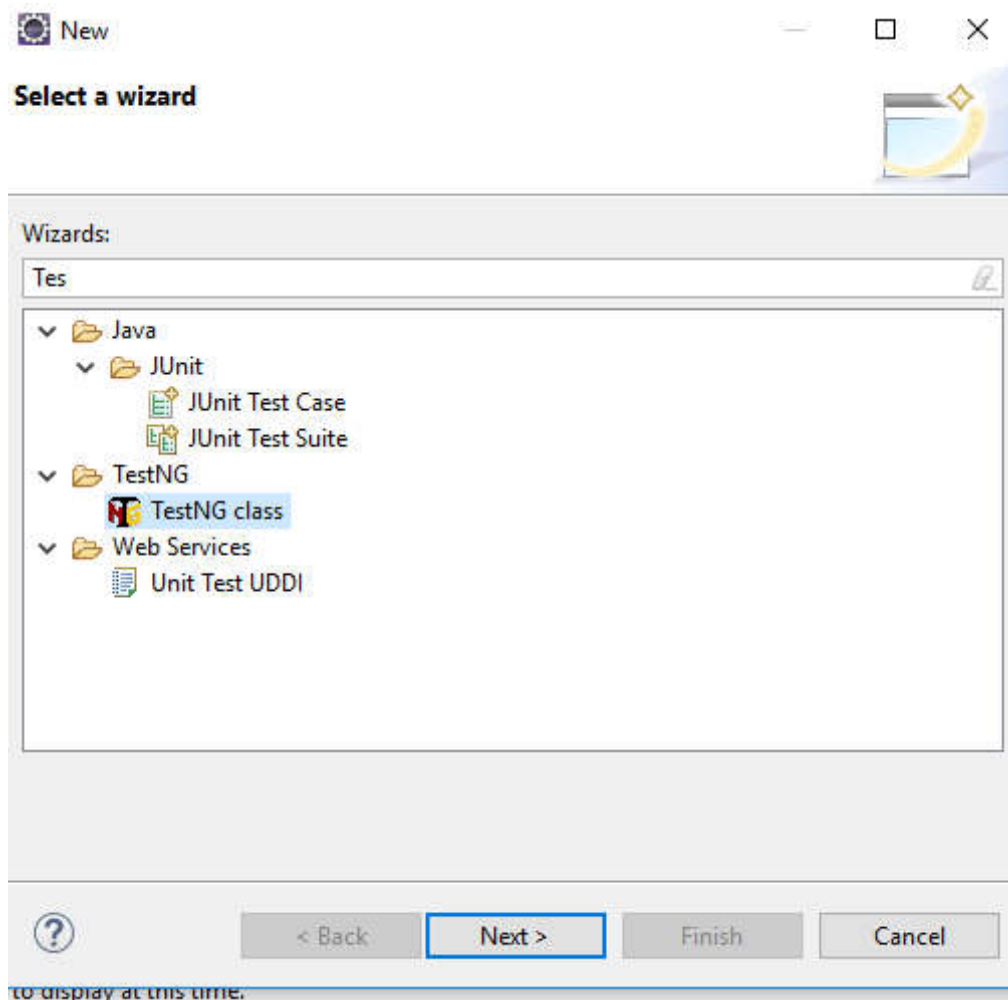
7. Project created successfully.

8. Add TestNG Plugin in Eclipse.

Install from update site

- Select *Help / Install New Software...*
- Enter the update site URL in “Work with:” field:
- Update site for release: <http://beust.com/eclipse>.
- Or, Update site for beta: <http://beust.org/eclipse-beta> , use it if you want to experiment with the new features or verify the bug fixes, and please report back if you encounter any issues.
- **Help → Install new software → Here uncheck “Contact all update sites during install to find required software”**
- Make sure the check box next to URL is checked and click *Next*.
- Eclipse will then guide you through the process.

10. Create a new TestNg Class



to display at this time.

NewTest.java

```
package example;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.firefox.FirefoxDriver;
```

```
import org.testng.Assert;

import org.testng.annotations.Test;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.AfterTest;

public class NewTest {

    private WebDriver driver;

    @Test

    public void testEasy() {

        driver.get("http://demo.guru99.com/test/guru99home/");

        String title = driver.getTitle();

        Assert.assertTrue(title.contains("Demo Guru99 Page"));

    }

    @BeforeTest

    public void beforeTest() {

        driver = new FirefoxDriver();

    }

    @AfterTest

    public void afterTest() {

        driver.quit();

    }

}
```

Right-click on the WebdriverTest and select **TestNG | Convert to TestNG**.
Eclipse will create testng.xml which says that you need to run only one test with the name **NewTest** as shown in the following screenshot:

The screenshot shows the TestNG configuration window. At the top, the checkbox 'Generate testng.xml' is checked. Below it, the 'Location' field contains '/WebdriverTest/testng.xml' with a 'Browse...' button to its right. The 'Suite name' field contains 'Suite' and the 'Test name' field contains 'Test'. Under 'Class selection', the 'Classes' dropdown is selected. 'Parallel mode' is set to 'none' and 'Thread count' is an empty field. The 'Preview' section shows the generated XML code. At the bottom, the 'Code generation' section has a 'suite() methods:' label and a 'Remove' dropdown.

☒ Generate testng.xml

Location:

Suite name:

Test name:

Class selection: Parallel mode: Thread count:

Preview

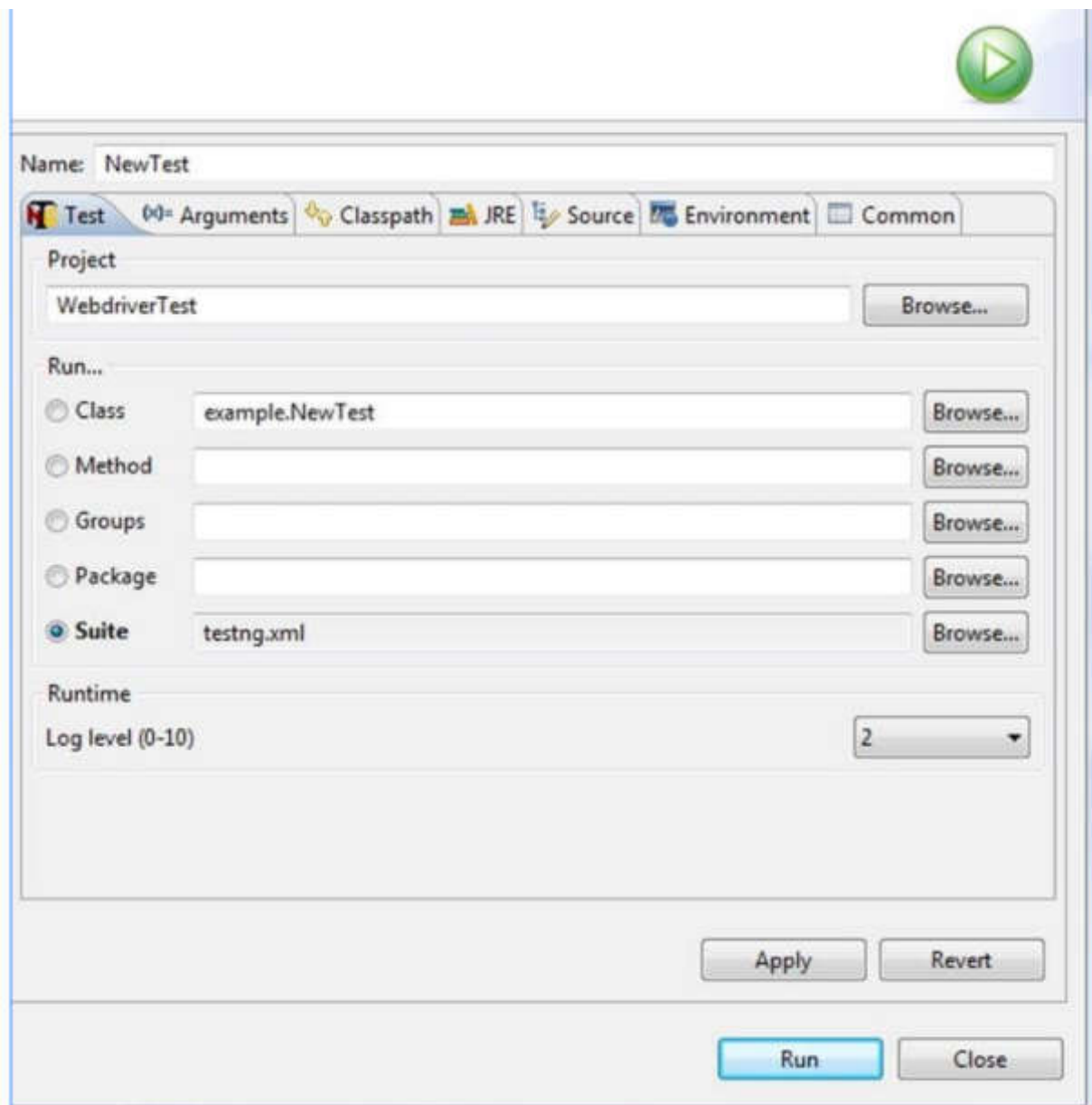
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite" parallel="none">
  <test name="Test">
    <classes>
      <class name="example.NewTest"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

Code generation

suite() methods:

Now you need to run test through this **testng.xml**.

So, go to the **Run Configurations** and create a new launch **TestNG**, select the project and field **Suite** as **testng.xml** and click Run



Build finished successfully

Now run as maven test.