

Testing web apps in the cloud (BrowserStack) using Selenium and TestNG in Java

Overview

In this tutorial, we will create a test in TestNG in order to validate a simple browser interaction using BrowserStack for cloud testing.



Please note

Within this tutorial, only one Test Execution will be used; it will contain one Test Run with all the results for the different used browsers. Thus, the overall test run status will be affected by the results made for all the browsers.

Instead of this approach, a different one could be creating a Test Execution per each browser; this would require some adaptations in order to produce a XML report per each used browser. This approach would give the ability to take advantage of Test Environments (more info in [Working with Test Environments](#)).

Requirements

- Install Java
- Install all dependencies using "mvn"

Description

This tutorial is based on [BrowserStack's own tutorial for TestNG](#).

You may start by cloning the repository <https://github.com/browserstack/testng-browserstack>.

```
git clone https://github.com/browserstack/testng-browserstack
```

The test, which validates a simple search using Google, is implemented in a SingleTest class which extends BrowserStackTestNGTest helper class. The latter provides setup related logic, including loading BrowserStack credentials from a configuration file.

src/test/java/com/browserstack/SingleTest.java

```
package com.browserstack;

import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;

import org.testng.Assert;
import org.testng.annotations.Test;

public class SingleTest extends BrowserStackTestNGTest {

    @Test
    public void test() throws Exception {
        driver.get("https://www.google.com/ncr");
        WebElement element = driver.findElement(By.name("q"));
        element.sendKeys("BrowserStack");
        element.submit();
        Thread.sleep(5000);

        Assert.assertEquals("BrowserStack - Google Search", driver.getTitle());
    }
}
```



Please note

If you wish to map the test method to an existing Test issue, or to create links to requirements whenever importing the results, please have a look at the tutorial [Testing using TestNG in Java](#) which provides the necessary instructions in order to setup TestNG for this purpose.

You need to configure the BrowserStack user/key along with desired browser capabilities/devices. We need to update the `config/single.conf.json` and `config/parallel.conf.json` files.

config/parallel.conf.json

```
{
  "server": "hub-cloud.browserstack.com",
  "user": "<user>",
  "key": "<key>",

  "capabilities": {
    "browserstack.debug": true
  },

  "environments": {
    "chrome": {
      "browser": "chrome"
    },
    "firefox": {
      "browser": "firefox"
    },
    "safari": {
      "browser": "safari"
    },
    "ie": {
      "browser": "internet explorer"
    }
  }
}
```

Test(s) can be run in parallel using the maven "parallel" task.

```
mvn test -P parallel
```

After successfully running the tests and generating the TestNG XML report (e.g. [testng-results.xml](#)), it can be imported to Xray (either by the REST API or through the **Import Execution Results** action within the Test Execution).

Overall Execution Status

3PASS

TOTAL TESTS: 3

FILTERS

Test Set

Assignee

Status

Component

Search

All

All

Contains text

X Clear

Showing 10 entries

Columns

	Key	Summary	Test Type	#Req	#Def	Test Sets	Assignee	Status
<div><div></div><div></div></div>	1	CALC-1938	test	Generic	0	0	Administrator	PASS
<div><div></div><div></div></div>	2	CALC-1939	tearDown	Generic	0	0	Administrator	PASS
<div><div></div><div></div></div>	3	CALC-1940	setUp	Generic	0	0	Administrator	PASS

Showing 1 to 3 of 3 entries

First Previous 1 Next Last

TestNG's tests are mapped to Generic Tests in Jira, and the **Generic Test Definition** field contains the namespace, the name of the class, and the method name that implements the Test case.

The execution screen details will not only provide information on the overall test run result, but also on a "browser" basis.

For each browser, a different "context" will appear along with the respective result.

Calculator / Test Execution: CALC-1937 / Test: CALC-1938
test

Export Test as Text

Return to Test Execution

Next

Execution Details

Test Description

None

Test Details

Test Type:

Generic


Definition:

com.browserstack.SingleTest.test

Results

Context	Error Message	Duration	Status
Parallel - SingleTestIE	-	10 sec	PASS
Parallel - SingleTestChrome	-	7 sec	PASS
Parallel - SingleTestFirefox	-	6 sec	PASS
Parallel - SingleTestSafari	-	6 sec	PASS

In BrowserStack you can see some info about it.

 **BrowserStack Automate**

[Buy a plan](#)

[Get help](#)

[Invite my team](#)

[Products](#)

[Account](#)

Free plan
85 mins left
[Invite my team](#)

[Buy a Plan](#)

Username and Access Keys [Show +](#)

Parallel threads
0/5 Running 0/5 Queued

All Projects
☒ Completed ☒ Timeout ☒ Errors

Build: testng browserstack 19
1m ago

Build: junit browserstack 5
5d ago

Check out our New REST API, which now includes the ability to update your Project, Build and Session names. [Find out more](#)

Build: testng browserstack

[Delete Build](#)

Build ID e3236fbd37eb452aa623c494e5e5e70e651ca82d

Started 18:21 UTC 7 Jun 2018

Duration 3 days 15 hrs 16 mins 53 secs

All Sessions (19) Completed (19) Timeouts (0) Errors (0)

Session	OS	Browser / Device	Duration	Finished
✓ parallel_test	Win 7	Chrome 67.0	25s	1m ago
✓ parallel_test	Win 10	IE 11.0	20s	1m ago
✓ parallel_test	Win 10	Firefox 60.0	19s	1m ago
✓ parallel_test	High Sierra	Safari 11.1	39s	1m ago

References

- <https://www.browserstack.com/automate/testng>
- <https://github.com/browserstack/testng-browserstack>