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 adaptions 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 later provides setup related logic, including loading BrowserStack credentials from a configuraiton 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.

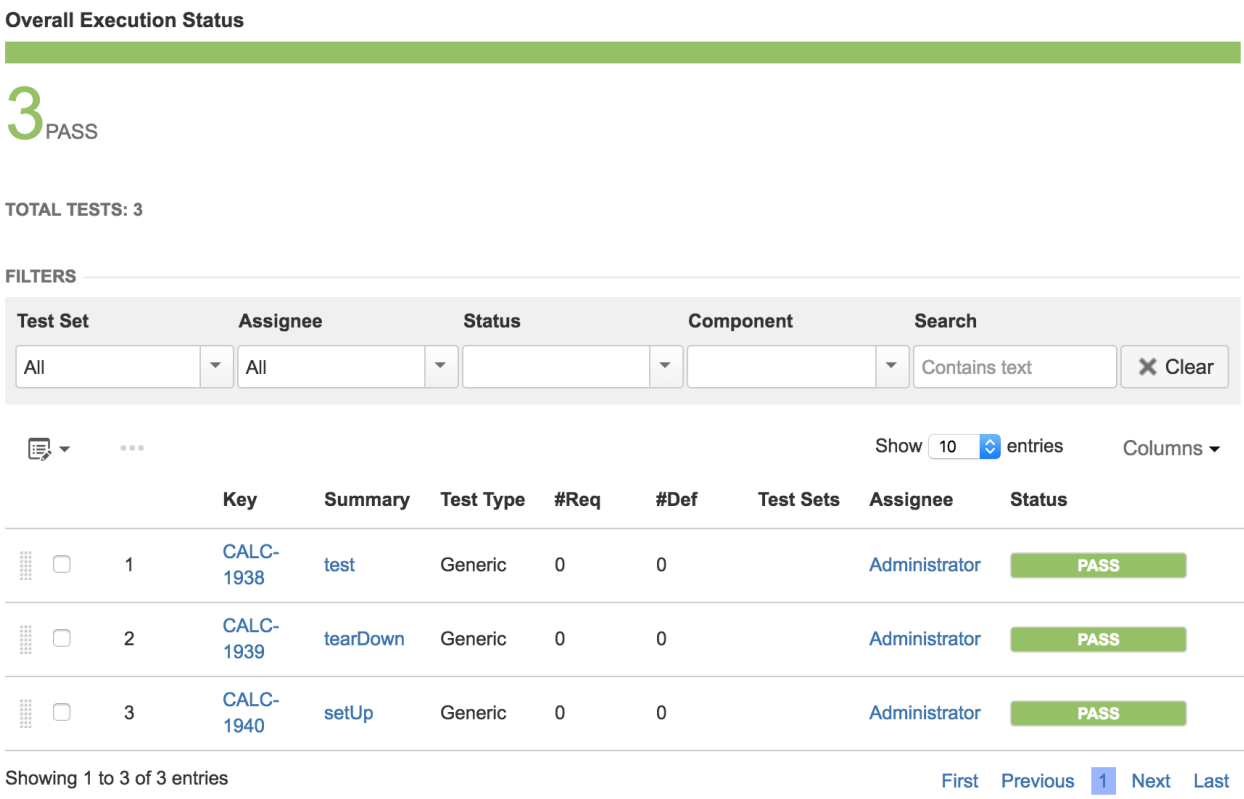
config/parallel.conf.json

```
{
  "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
```

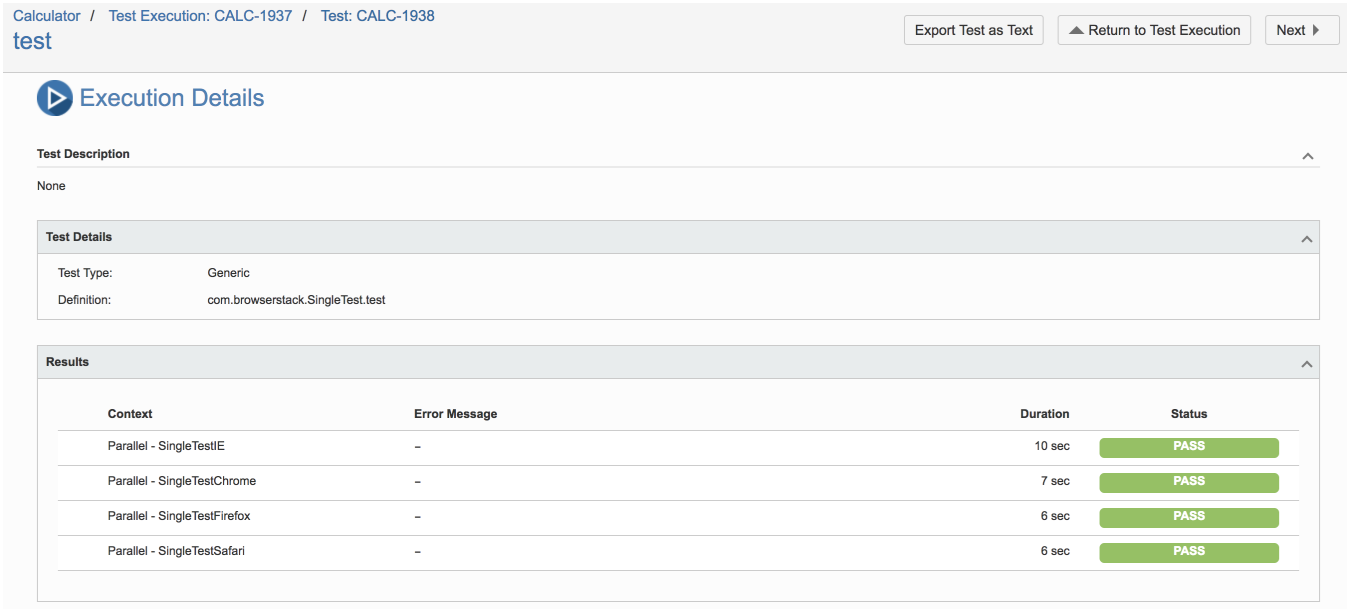
After successfully running the tests and generating the TestNG XML report (e.g. testng-results.xml), it can be imported to Xray (either by the REST API or through the **Import Execution Results** action within the Test Execution).



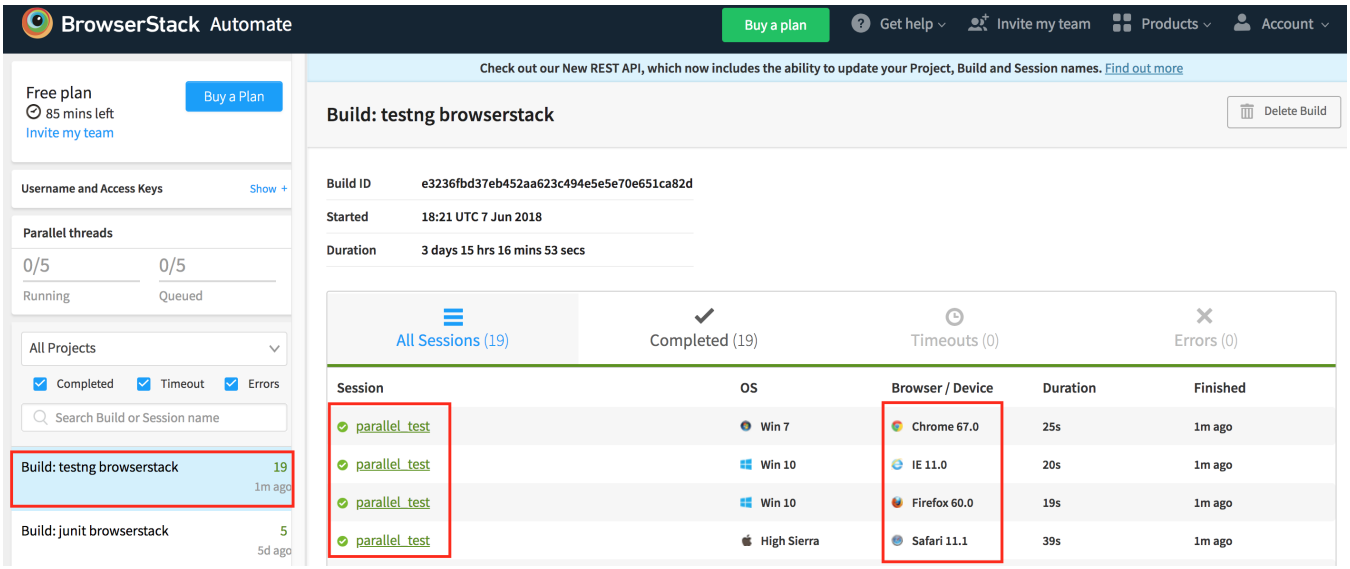
TestNG's tests are mapped to Generic Tests in Jira, and the **Generic Test Definition** field contains the namespace, the name of the class, and the method name that implements the Test case.

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.



In BrowserStack you can see some info about it.



References

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