

BrowserStack Plugin

Plugin Information

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

This plugin allows you to integrate with [BrowserStack](#).

Overview

BrowserStack is the industry's most reliable cloud web and mobile testing platform, used by more than 2 million developers across 135 countries. BrowserStack gives instant access to 2000+ real mobile devices and browsers that enables developers to test their websites and mobile applications without requiring to install or maintain an internal lab of virtual machines, devices or emulators. To start using BrowserStack, [signup for a free trial](#) account.

More than 25,000 global companies rely on BrowserStack, including industry-leaders like Microsoft, ING, Mastercard, Dow Jones, Garmin, National Geographic, Volvo, NRK, HubSpot, Walt Disney and AngularJS, to name a few. With offices in San Francisco, Mumbai and Dublin, BrowserStack is privately held and backed by Accel. Learn more at: www.browserstack.com.

Features

1. Manage your BrowserStack credentials globally or per build job.
2. Set up and tear down [BrowserStack Local](#) for testing internal, dev or staging environments.
3. Embed BrowserStack [Automate](#) or BrowserStack [App Automate](#) reports in your Jenkins job results.

Requirements

1. Minimum Jenkins version supported is 1.653+
2. If you are using Java, we support all versions of Java up to Java 9
3. For viewing BrowserStack Automate report in your Jenkins test results.
 1. Jenkin's [JUnit plugin](#) must be installed.
 2. We support Java with JUnit and TestNG
 3. We support Node with Protractor and WebdriverIO

Configuring tests

This step is required for viewing the BrowserStack Report in your Jenkins job results.

Java - TestNG and JUnit

Java projects must be built with Maven and use either TestNG or JUnit. Add the following dependencies to your pom.xml file:

```
<dependency>
  <groupId>com.browserstack</groupId>
  <artifactId>automate-testassist</artifactId>
  <version>1.0.0-SNAPSHOT</version>
</dependency>
```

Add the following repositories to your pom.xml file:

```
<repositories>
  <repository>
    <id>sonatype-nexus-snapshots</id>
    <url>https://oss.sonatype.org/content/repositories/snapshots</url>
  </repository>
</repositories>

<pluginRepositories>
  <pluginRepository>
    <id>sonatype-nexus-snapshots</id>
    <url>https://oss.sonatype.org/content/repositories/snapshots</url>
  </pluginRepository>
</pluginRepositories>
```

Add the following plugin to your pom.xml file:

```
<build>
  <plugins>
    <plugin>
      <groupId>com.browserstack</groupId>
      <artifactId>automate-maven-plugin</artifactId>
      <version>1.0.0-SNAPSHOT</version>
      <configuration>
        <source>${jdk.source.version}</source>
        <target>${jdk.target.version}</target>
        <complianceLevel>${jdk.source.version}</complianceLevel>
      </configuration>
      <executions>
        <execution>
          <goals>
            <goal>test-compile</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

You must define the maven properties, **jdk.source.version** and **jdk.target.version** to match the Java versions of your project.

WebdriverIO

To configure your project:

Add wdio-browserstack-reporter as a dependency in your package.json file.

package.json

```
{
  "dependencies": {
    "wdio-browserstack-reporter": "~1.0.0"
  }
}
```

Add browserstack as a reporter in your conf file.

conf.js

```
// wdio.conf.js
module.exports = {
  // ...
  reporters: ['browserstack'],
  reporterOptions: {
    browserstack: {
      outputDir: './'
    }
  },
  // ...
};
```

outputDir (optional): defines a directory location for storing browserstack report files.

[Here is the link for the GitHub repository.](#)

Please note that WebdriverIO support works with Automate and not with App-Automate right now.

Protractor

To configure your project:

Add protractor-browserstack-reporter as a dependency in your package.json file.

package.json

```
{
  "dependencies": {
    "protractor-browserstack-reporter": "~1.0.0"
  }
}
```

Add browserstack as a plugin in your conf file.

conf.js

```
// conf.js
'plugins': [{
  'package': "browserstack-protractor-reporter"
}]
```

outputDir (optional): defines a directory location for storing browserstack report files.

[Here is the link for the GitHub repository.](#)

Please note that Protractor support works with Automate and not with App-Automate right now.

Following environment variables are set by this Jenkins plugin:

```
BROWSERSTACK_USER  
BROWSERSTACK_ACCESSKEY  
BROWSERSTACK_LOCAL  
BROWSERSTACK_LOCAL_IDENTIFIER
```

You can use these environment variables to set DesiredCapabilities in your test. For e.g:

```
String username = System.getenv("BROWSERSTACK_USER");  
String accessKey = System.getenv("BROWSERSTACK_ACCESSKEY");  
String browserstackLocal = System.getenv("BROWSERSTACK_LOCAL");  
String browserstackLocalIdentifier = System.getenv(  
    "BROWSERSTACK_LOCAL_IDENTIFIER");  
  
DesiredCapabilities capabilities = new DesiredCapabilities();  
capabilities.setCapability("os", "Windows");  
capabilities.setCapability("browser", "chrome");  
capabilities.setCapability("browserstack.local", browserstackLocal);  
capabilities.setCapability("browserstack.localIdentifier",  
    browserstackLocalIdentifier);  
driver = new RemoteWebDriver(new URL("https://" + username + ":" +  
    accessKey + "@hub.browserstack.com/wd/hub"), capabilities);
```

Global Configuration

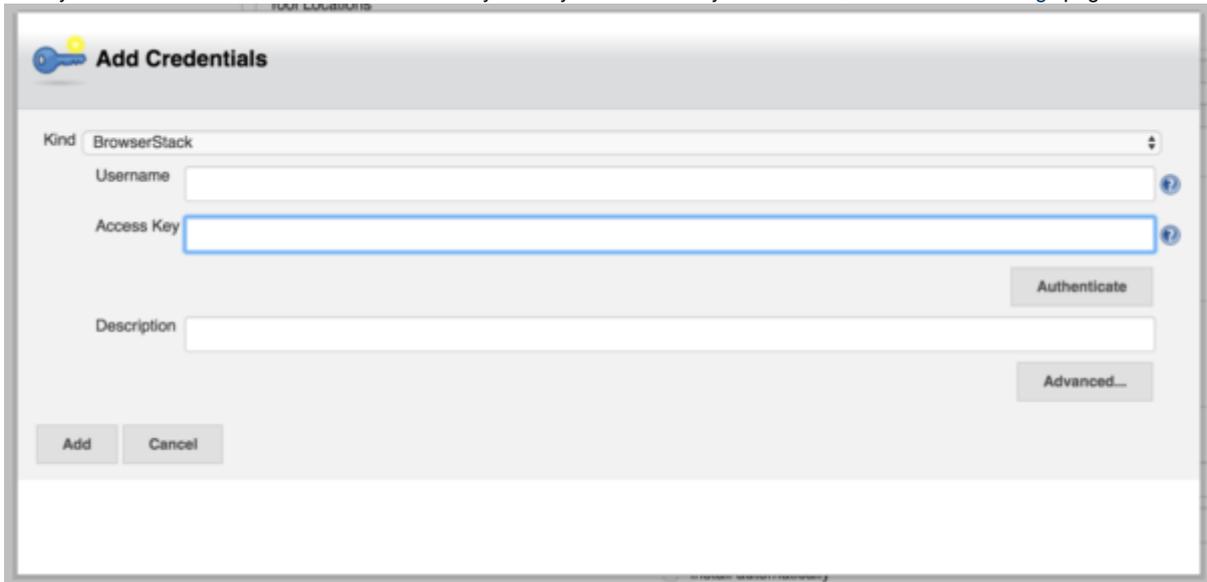
BrowserStack Credentials

To configure BrowserStack credentials globally follow these steps,

1. Go to **Manage Jenkins > Configure System**.
2. Under the **BrowserStack** section click on the **Add** button next to **BrowserStack Credentials**.
 1. In the global section, there is an additional option, that if checked allows the plugin to collect anonymous usage data.



3. Enter your BrowserStack username and access key which you can find on your BrowserStack [Account Settings](#) page.



The screenshot shows the 'Add Credentials' dialog box. At the top, there is a key icon and the title 'Add Credentials'. Below the title, there is a 'Kind' dropdown menu currently set to 'BrowserStack'. Underneath, there are two text input fields: 'Username' and 'Access Key'. The 'Access Key' field is highlighted with a blue border. To the right of the 'Access Key' field is a question mark icon. Below the input fields, there are two buttons: 'Authenticate' and 'Advanced...'. At the bottom left, there are two buttons: 'Add' and 'Cancel'.

4. Save your changes.

Job Configuration

Create a new Jenkins job or edit the configuration of an existing one. Before you can edit/add job configuration you will need to ensure that you have the necessary privileges on your Jenkins instance to add/modify jobs.

BrowserStack Credentials

1. Under the **Build Environment** section check the box next to BrowserStack.



The screenshot shows the 'Build Environment' section of a Jenkins job configuration. There are two checkboxes: 'BrowserStack' (checked) and 'BrowserStack Local' (unchecked). Below the 'BrowserStack' checkbox, there is a 'BrowserStack Credentials' dropdown menu and an 'Add' button. Below the 'BrowserStack Local' checkbox, there is a note: 'Note: Skip this if you are using BrowserStack Local bindings in your code.'

2. If you have previously saved any global credentials these will be automatically selected.
3. If you have not previously saved any global credentials or wish to override credentials for this job you can do so by clicking on Add.
4. The **BrowserStack Credentials** screen will show up again where you can enter your BrowserStack username and access key.

BrowserStack Local

[BrowserStack Local Testing](#) allows you to test your private and internal servers, alongside public URLs, using the BrowserStack cloud, which has support for firewalls, proxies and Active Directory. The plugin is responsible for downloading the binary for the platform that the build job is running on and starting and tearing down the secure tunnel.

BrowserStack Local can only be configured per job and not via the global configuration.

To enable BrowserStack Local follow these steps,

1. Under the **Build Environment** section check the box next to **BrowserStack Local**.



The screenshot shows the 'Build Environment' section of a Jenkins job configuration. There are two checkboxes: 'BrowserStack' (checked) and 'BrowserStack Local' (checked). Below the 'BrowserStack' checkbox, there is a 'BrowserStack Credentials' dropdown menu and an 'Add' button. Below the 'BrowserStack Local' checkbox, there are two text input fields: 'BrowserStack Local Path' and 'BrowserStack Local Options'. Below the input fields, there is a note: 'Note: Skip this if you are using BrowserStack Local bindings in your code.'

2. If the binary is externally downloaded then set the path to the binary in the config path, **BrowserStack Local Path**.
3. Downloading the binary is not necessary and if this option is left empty the plugin will automatically download the binary. This is recommended especially if you are using Jenkin's in master-slave configurations, since the plugin will download the appropriate binary for the build agent OS.
4. You can set additional options for the binary in the [BrowserStack Local Options](#).

Embedded Test Reports

To have BrowserStack Automate reports show up right next to the JUnit/TestNG reports in Jenkins you must follow these steps,

1. Click on **Add post-build Action** under the Post-build Actions section.



2. Click on **Additional test report features**.



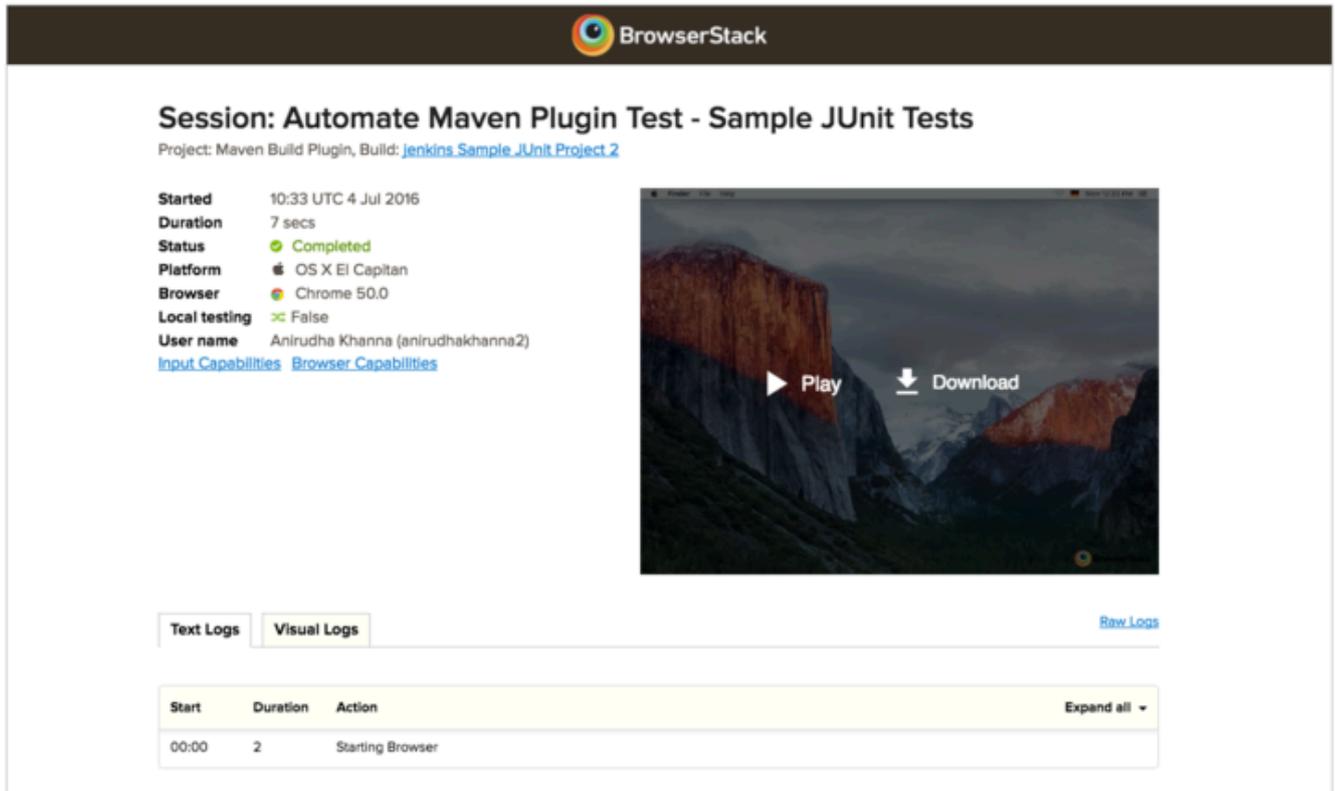
3. Check the box next to **Embed BrowserStack Report**.



DONE! This will ensure that once the build runs, the BrowserStack Automate report will show up right next to the stack trace of a failed unit test.

BrowserStack

[View this on BrowserStack Automate Dashboard](#)

A screenshot of a BrowserStack Automate report. The report title is 'Session: Automate Maven Plugin Test - Sample JUnit Tests'. Below the title, it shows the project name 'Project: Maven Build Plugin, Build: Jenkins_Sample_JUnit_Project_2'. The report details include: Started at 10:33 UTC 4 Jul 2016, Duration of 7 secs, Status of 'Completed', Platform of 'OS X El Capitan', Browser of 'Chrome 50.0', Local testing of 'False', and User name of 'Anirudha Khanna (anirudhakhanna2)'. There are links for 'Input Capabilities' and 'Browser Capabilities'. A video player shows a screenshot of a mountain landscape with 'Play' and 'Download' buttons. Below the video, there are tabs for 'Text Logs' and 'Visual Logs', and a 'Raw Logs' link. At the bottom, there is a table with columns 'Start', 'Duration', and 'Action'.

Start	Duration	Action	Expand all
00:00	2	Starting Browser	

App Automate support

With the plugin version 1.1.0 rollout we have added support for App Automate in Jenkins plugin. In addition to all the above configurations, you will now be able to add a build step to configure and upload your app on the BrowserStack servers. Refer to the [App Automate Jenkins](#) document for details on configuring your test suite for automation app testing. Visit [App Automate Get Started](#) document for more details on getting started with the automation app testing on BrowserStack.

Change Log

Version 1.1.0 (May 09 2018)

- App Automate support.
- Version upgrades for automate-testassist and automate-maven-plugin to 1.0.0-SNAPSHOT to fix critical bugs.
 - Support for Java 9 in automate-maven-plugin
 - Fixes for bug in capturing report when using @Parameters or @DataProvider annotations in JUnit or TestNG

Version 1.0.7 (Nov 03 2016)

- Fixed security bug.

Version 1.0.3 (Jul 29 2016)

- Minor fix to analytics.

Version 1.0.2 (Jul 14 2016)

- Fixed broken url in pom.xml for plugin wiki.

Version 1.0.1 (Jul 13 2016)

- Fixed typos and pom.xml.

Version 1.0.0 (Jul 12 2016)

- Initial release of plugin.