

T  
E  
S  
T  
E  
R  
S  
  
Z  
O  
N  
E

# BrowserStack CrossBrowserTesting Platform



Mithilesh Singh



# What is Browser Stack?

- Browser Stack is an cloud web and mobile testing platform that provides developers with the ability to test their websites and mobile applications across on-demand browsers, operating systems and real mobile devices. They have following primary products:

Test your websites

## LIVE

Interactive cross browser testing



## AUTOMATE

Selenium testing at scale



## PERCY

Visual testing and review



Test your mobile apps

## APP LIVE

Interactive mobile app testing



## APP AUTOMATE

Automated mobile app testing



# Why Browser Stack?

- 
1. we can avoid the hassle of having to buy all the mobile devices that are available today to perform the testing.
  2. We can use Browser Stack as a remote lab, and we can even use it as Real Desktop Browsers or Mobile Browsers.
  3. There is no setup required for using Browser Stack.
  4. We can use it directly on any independent machine by using the Browser Stack URL and its login credentials

## Types of Live Testing



- Functionality Testing
- Usability testing
- Interface testing
- Compatibility testing
- Performance testing
- Security testing

These Kind of testing we can do on the browser stack platform.



How to  
connect  
with  
Browser  
Stack ?

How to  
connect  
with  
Browser  
Stack ?

- Access the Browser stack URL: <https://www.browserstack.com/>

The screenshot shows the homepage of browserstack.com. At the top, there's a navigation bar with links for Products, Developers, Live for Teams, Pricing, Sign in, and a FREE TRIAL button. Below the header, a large title reads "App & Browser Testing Made Easy". A subtext message encourages users to "Give your users a seamless experience by testing on 3000+ real devices and browsers. Don't compromise with emulators and simulators." Two prominent buttons are visible: "Get started free" (in blue) and "Get a demo". The main content area is divided into two sections: "Test your websites" and "Test your mobile apps". Each section contains three cards, each with an icon and a brief description:

- LIVE**: Interactive cross browser testing. Illustration shows a person interacting with multiple browser windows.
- AUTOMATE**: Selenium testing at scale. Illustration shows a person working on a laptop with code snippets like "php" and "python".
- PERCY**: Visual testing and review. Illustration shows a person holding a smartphone and a tablet, with a gear icon.
- APP LIVE**: Interactive mobile app testing. Illustration shows a person interacting with a mobile device.
- APP AUTOMATE**: Automated mobile app testing. Illustration shows a person sitting cross-legged, surrounded by mobile devices and icons.

This attachment is as per latest browser stack UI, it might get modified in future.

LIVE

Interactive cross browser testing



# Available OS for the live testing.

## Windows



Windows 11 (64-bit)



Windows 10 (64-bit)



Windows 8.1 (64-bit)



Windows 8 (64-bit)

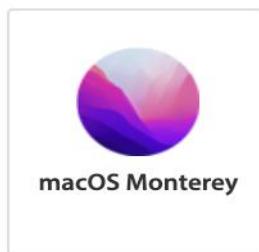


Windows 7 (64-bit)



Windows XP (32-bit)

## Mac



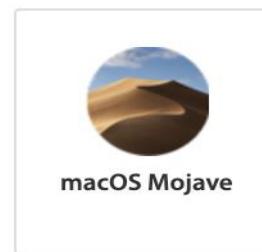
macOS Monterey



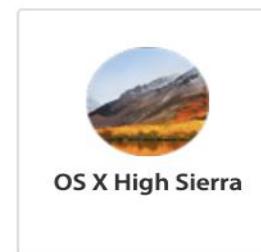
macOS Big Sur



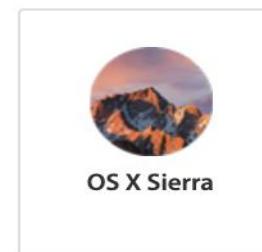
macOS Catalina



macOS Mojave



OS X High Sierra



OS X Sierra



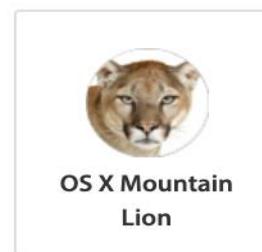
OS X El Capitan



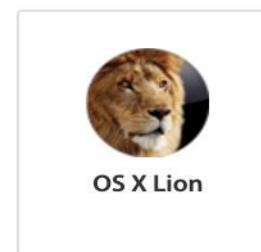
OS X Yosemite



OS X Mavericks



OS X Mountain Lion



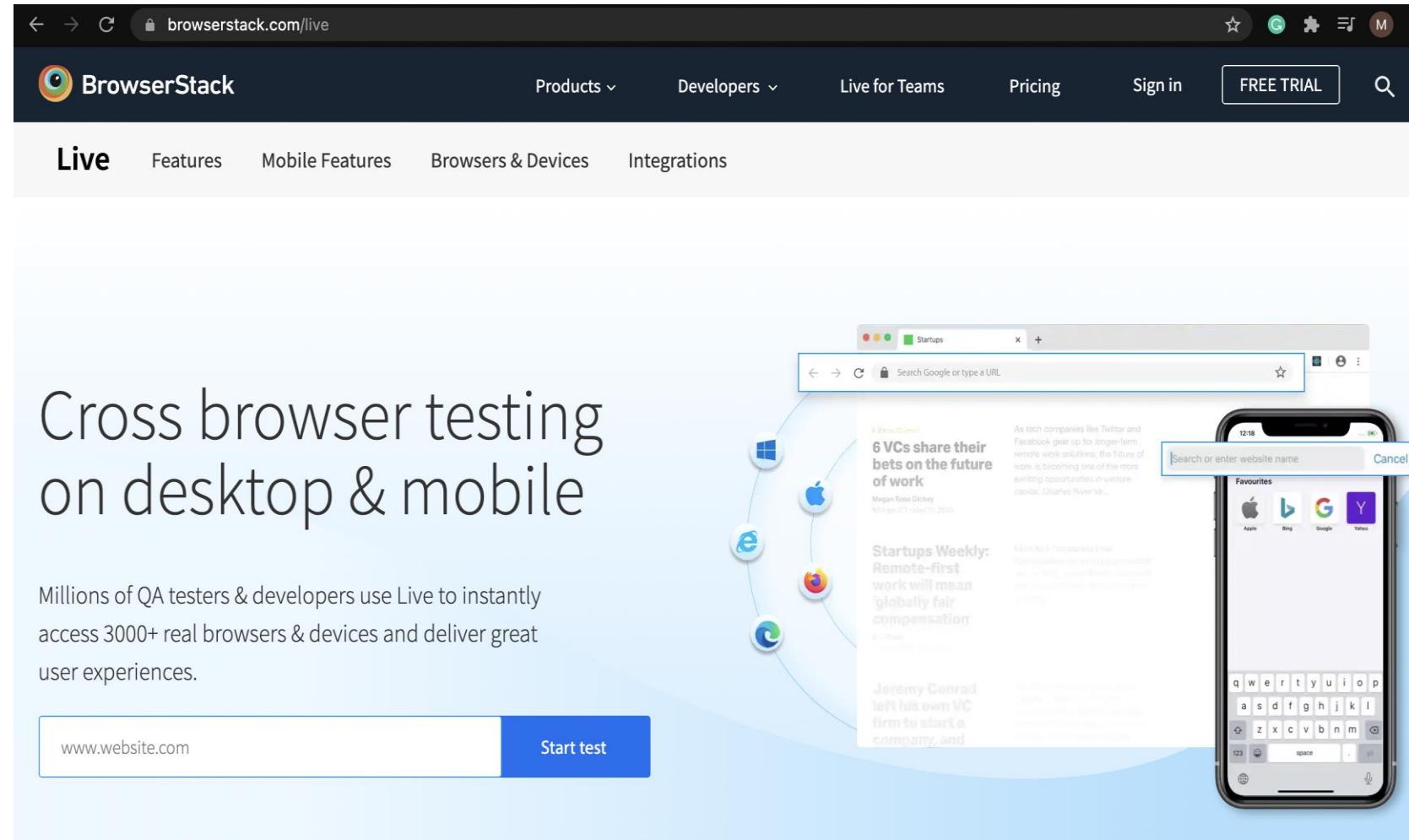
OS X Lion



OS X Snow Leopard

## 2. Choose Live option

We have **Live**,  
**Automate** and **Percy**  
options for websites  
testing as per current  
UI.



The screenshot shows the BrowserStack Live testing interface. At the top, there's a navigation bar with links for Products, Developers, Live for Teams, Pricing, Sign in, and a FREE TRIAL button. Below the navigation, there are tabs for Live, Features, Mobile Features, Browsers & Devices, and Integrations. The main content area features a large heading "Cross browser testing on desktop & mobile". Below this, a paragraph states: "Millions of QA testers & developers use Live to instantly access 3000+ real browsers & devices and deliver great user experiences." To the right, there's a preview window showing a desktop browser (Chrome) and a mobile phone displaying a news article from Startups Weekly. The desktop browser has a search bar with "Search Google or type a URL" and a favourites bar with icons for Apple, Bing, Google, and Yahoo. The mobile phone screen shows a news article with a headline about tech companies like Twitter and Facebook preparing for long-term remote work solutions.

3. enter the testing URL and click on Start Test

The screenshot shows the BrowserStack Live interface. At the top, there's a navigation bar with links for Products, Developers, Live for Teams, Pricing, Sign in, and a FREE TRIAL button. Below the navigation bar, there are tabs for Live, Features, Mobile Features, Browsers & Devices, and Integrations. The main content area features a large heading "Cross browser testing on desktop & mobile" with a subtext explaining that millions of QA testers & developers use Live to instantly access 3000+ real browsers & devices. A search bar at the bottom left contains the URL "http://zero.webappsecurity.com/login.html". To the right of the search bar is a blue "Start test" button. In the center, there's a diagram showing five browser icons (Windows, Apple, Edge, Firefox, and Chrome) connected by lines, symbolizing cross-browser compatibility. To the right of the diagram, there's a screenshot of a desktop browser window displaying news articles, and below it, a screenshot of an iPhone displaying a search interface.

4. Choose OS(Window or Mac) and Browser(chrome, Firefox, IE etc.) for Web testing or Real devices for mobile testing.

 **BrowserStack** Live

Products ▾  Invite my team  Plans and pricing  Get help ▾   1 

 Each device is available for up to 1 minute during Free Trial. For full access: [Buy a plan](#)

Quick Launch							
 Android	95 Latest	11 Latest	94 Latest	96 Latest	81 Latest	14.12 Latest	5.1 Latest
 iOS	96 Beta		95 Beta	97 Beta	82 Dev		
 Windows 11	97 Dev		93	98 Dev	80		
 10	94		92	95	79		
	93		91	94	78		
	92		90	93	77		
	91		89	92	76		
 7	90		88	91	75		
 XP	89		87	90	74		
 Mac	88		86	89	73		
+ 	11 more		54 more	51 more	48 more		

Drag a browser here to add to Quick Launch

Login page is appeared after accessing the URL.

The screenshot shows a web browser window with two tabs. The top tab is titled "Zero - Log in" and the bottom tab is titled "Not secure | zero.webappsecurity.com/login.html". The main content area displays a login form for "Zero Bank" with the title "Log in to ZeroBank". The form includes fields for "Login" and "Password", a "Keep me signed in" checkbox, and a "Sign in" button. Below the form is a link "Forgot your password ?". To the left of the main content is a sidebar menu with the following options: "Switch browser", "Local Testing", "1307 x 724", "Report a bug" (which has a red arrow pointing to it), "Settings", "Change Location", "Screen Reader", and "Stop Session". At the bottom of the page, there are links for "Download WebInspect", "Terms of Use", "Contact Micro Focus", and "Privacy Statement".

We can report any issues to JIRA from the Browser stack testing platform after clicking on report a bug(observe in previous slide).

The screenshot shows a web browser window with the URL `live.browserstack.com/dashboard?try_live_url=http%3A%2F%2Fzero.webappsecurity.com%2Flogin.html#os=Windows&os_version=11&browser=C...`. The main content area displays a "Zero - Log in" page titled "Log in to ZeroBank". It features input fields for "Login" and "Password", a "Keep me signed in" checkbox, and a "Sign in" button. Below the form is a link "Forgot your password ?". At the bottom of the page, there are links for "Download Webinspect", "Terms of Use", "Contact Micro Focus", and "Privacy Statement". A note at the bottom states: "The Free Online Bank Web site is published by Micro Focus Fortify for the sole purpose of demonstrating the functionality and effectiveness of Micro Focus Fortify's WebInspect products in detecting and reporting Web application".

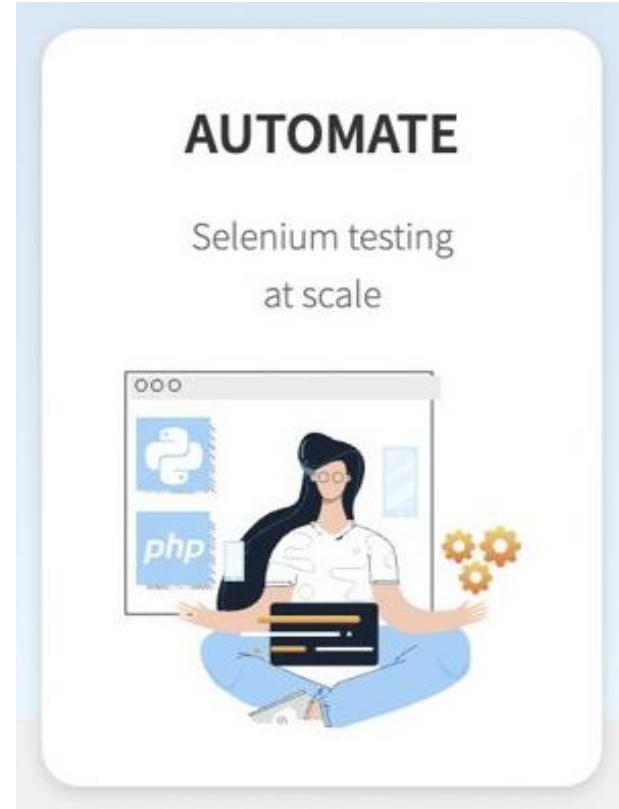
The right side of the screen shows a "REPORT AN ISSUE" sidebar. It has a title "One Click Bug Filing with the following details pre-filled:" followed by a clipboard icon. Below this is a section titled "Select tool" with a list of options:

- Email
- Download
- Jira
- Trello
- Github
- Slack

At the bottom of the slide, there is a footer bar with icons for tests passed (1), a date (yesterday 7:59 PM), and other navigation links.

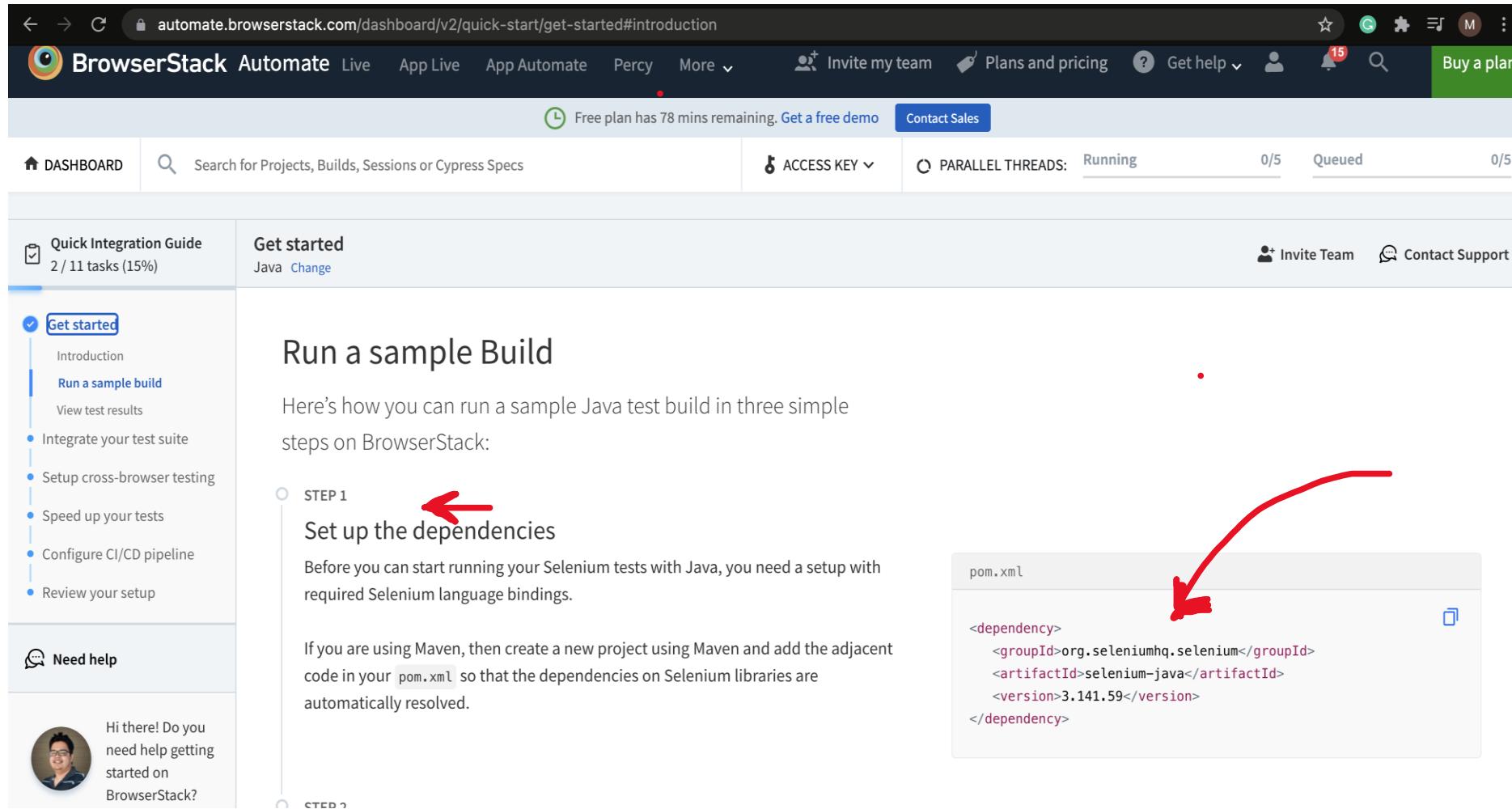
Since we are using trial version we will be getting 1 min of access of each browser. If we want some more time then need to purchase a plan. This screen will appear once we will try to use same browser version which is already used once.

The screenshot shows the BrowserStack Live interface. At the top, there's a navigation bar with the BrowserStack logo, 'Products', 'Invite my team', 'Plans and pricing', 'Get help', user profile, and a search icon. On the left, a sidebar lists 'Quick Launch', 'Android', 'iOS', 'Windows' (with options for 11, 10, 8.1, 8, 7, and XP), and 'Mac'. A modal window is centered over the interface, featuring a background image of a smartphone, tablet, and desktop monitor. The text inside the modal reads: 'You have already used the available Free Trial minutes of this browser. Upgrade now to enjoy unrestricted testing.' Below this is a green 'Upgrade' button and a 'No thanks' link. At the bottom of the modal, it says 'Drag a browser here to add to Quick Launch'. The background of the interface shows a grid of browser icons for different platforms and versions, such as '14.12 Latest' and '5.1 Latest'.



Browser Stack Automate option helps to run the automation script on the browser stack platform. We can select the specific browser and OS where we want to run the automated scripts. That will provide us snippet of code which we can copy and paste in our framework.

Click on Automate option available right to Browser Stack logo and follow the Steps to get the snippet of codes.



The screenshot shows the BrowserStack Automate dashboard with the 'Get started' guide open. A red arrow points to the 'Set up the dependencies' step, and another red arrow points to the `pom.xml` code snippet, specifically highlighting the Selenium dependency section.

**Quick Integration Guide**  
2 / 11 tasks (15%)

**Get started**  
Java Change

**Run a sample Build**

Here's how you can run a sample Java test build in three simple steps on BrowserStack:

**STEP 1** Set up the dependencies

Before you can start running your Selenium tests with Java, you need a setup with required Selenium language bindings.

If you are using Maven, then create a new project using Maven and add the adjacent code in your `pom.xml` so that the dependencies on Selenium libraries are automatically resolved.

```
pom.xml
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.141.59</version>
</dependency>
```

Select the test OS and Browser and then copy the code and paste inside your framework.

STEP 2   
Set up your first build

a. Configure the browser or device combinations

Choose any three desktop browser or devices from the thousands of combinations available at BrowserStack:

# 1	OS X Big Sur	chrome 92.0
# 2	OS X Big Sur	safari 14.1
# 3	OS X Big Sur	Safari 14.1

b. Copy the code 

Copy the code snippet on the right and save it in a file on your local computer. In the next step, we will run this script using Maven.



```
first-sample-build.java   
import java.util.concurrent.TimeUnit;  
import org.openqa.selenium.By;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.remote.DesiredCapabilities;  
import org.openqa.selenium.remote.RemoteWebDriver;  
import org.openqa.selenium.support.ui.ExpectedConditions;  
import org.openqa.selenium.support.ui.WebDriverWait;  
class TestClass1 implements Runnable {  
    public void run() {  
        Hashtable<String, String> capsHashtable = new Hashtable<  
        capsHashtable.put("browser", "chrome");  
        capsHashtable.put("browser_version", "92.0");  
        capsHashtable.put("os", "OS X");  
        capsHashtable.put("os_version", "Big Sur");  
        capsHashtable.put("build", "browserstack-build-1");  
        capsHashtable.put("name", "Thread 1");  
        mainTestClass r1 = new mainTestClass();  
        r1.executeTest(capsHashtable);  
    }  
    class TestClass2 implements Runnable {  
        public void run() {  
            Hashtable<String, String> capsHashtable = new Hashtable<
```

STEP 3 ←

## Execute build on BrowserStack

If you are using Maven, use the command shown on the right. If you are using Java IDEs like Eclipse or IntelliJ IDEA, you can save the entire code snippet as a class and run it directly from the IDE.

```
mvn test
```

- i If using IDE, download [Selenium Java bindings](#) and add it in the Classpath (for Eclipse) and Module Dependencies (for IntelliJ IDEA).

**Note:** Add this snippet of code and execute the automated scripts. Automation script will execute on the browser stack platform and execution details will be getting generated.



Observe The execution result on the Browser stack platform



← → C 🔒 automate.browserstack.com/dashboard/v2/builds/11f7499a2eb680d4da620cf4c989b2c606036fa4?overallStatus=completed ⭐ G M :

Quick Integration Guide  
2 / 11 tasks (15%)

Run a sample build  
[View test results](#)

- Integrate your test suite
- Setup cross-browser testing
- Speed up your tests
- Configure CI/CD pipeline
- Review your setup

**browserstack-build-1**

Sessions 1 SESSIONS Build Status COMPLETED  
Last Updated 05 Dec 2021 11:29 UTC User Mithilesh Singh  
Duration 18s Build ID 11f7499... [Copy](#)

≡ ALL SESSIONS (1) ✔ PASSED (1) ⚠ TIMED OUT (0) ❗ ERROR (0)

Sort Filter Search

Show Unmarked Duration Status ⓘ

**Welcome to the new session listing layout!**  
The new layout focuses on highlighting relevant assertions and debugging information — making it quicker and easier to debug tests.  
Here is a walkthrough of what's changed:

Quickly find OS and Browser/Device info beside the Session name

Simplified Session status combines assertion-based pass/fail status with execution status. Click on STATUS ⓘ for more info

FILTERS ▾ Build executed on the Browser stack platform Your session name here 2m 14s ● FAILED Session marked as failed!  
browserstack-build-1 1 SESSIONS • a few secs ago

Firefox 83 Win 10 • Last updated a few secs ago

**Note:** Browser stack generates the execution video recording which we can get on the same screen by scrolling window little bit down. Observe next slide.



# Browser stack execution summary in form of recording. We can expand it and get some more details.

automate.browserstack.com/dashboard/v2/builds/11f7499a2eb680d4da620cf4c989b2c606036fa4?overallStatus=completed

Quick Integration Guide  
2 / 11 tasks (15%)

Run a sample build  
[View test results](#)

- Integrate your test suite
- Setup cross-browser testing
- Speed up your tests
- Configure CI/CD pipeline
- Review your setup

Show Unmarked

ALL SESSIONS (1) PASSED (1) TIMED OUT (0) ERROR (0)

Sort Filter Search

Duration Status ⓘ

Here is a walkthrough of what's changed:

Quickly find OS and Browser/Device info beside the Session name

Your session name here

2m 14s • FAILED

Simplified Session status combines assertion-based pass/fail status with execution status. Click on STATUS ⓘ for more info

Session marked as failed!

View pass/fail reason without opening session details

Recording of the Execution

Thread 1 →

1 SESSIONS • a few secs ago

18s UNMARKED

Thread 1

Chrome 92.0 ⚡ Big Sur • Last updated 5 hrs ago

We can play and observe the recording, download the recording or can get different execution logs from here.



automate.browserstack.com/dashboard/v2/builds/11f7499a2eb680d4da620cf4c989b2c606036fa4/sessions/03b7ae99f60fbf00e1b5501e529912f5e48928e0?o... ☆ G M :

Quick Integration Guide  
2 / 11 tasks (15%)

- Run a sample build
- [View test results](#)
- Integrate your test suite
- Setup cross-browser testing
- Speed up your tests
- Configure CI/CD pipeline
- Review your setup

Need help

Hi there! Do you need help getting started on BrowserStack?

Talk to automation expert

FILTERS ▾

browserstack-build-1

1 SESSIONS • a few secs ago

Build > Thread 1

Browser: Chrome 92.0, OS: OS X Big Sur, Duration: 18s, Status: Unmarked

REST API, UNMARKED, Started At: 05 Dec 2021 11:28 UTC, Local Testing: Off, User: Mithilesh Singh

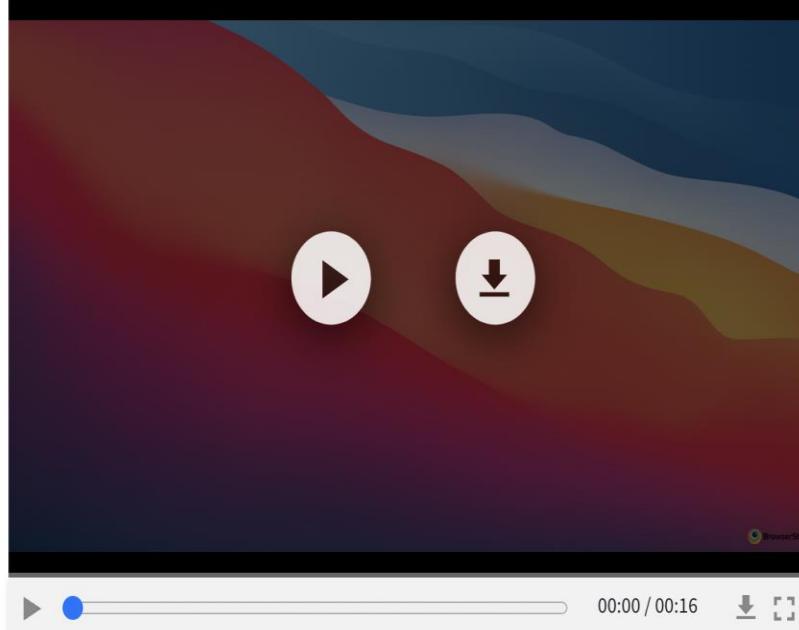
Session ID: 03b7ae99f60fbf00e1b5501e529912f5e48928e0, Public Link: [Copy Link](#)

Input Capabilities

os	OS X
build	browserstack-build-1
browser	chrome
os_version	Big Sur
name	Thread 1
browser_version	92.0
acceptSslCert	false
detected_language	selenium/3.141.59 (java mac)
realMobile	true
browserstack.seleniumLogs	true
browserstack.console	errors
browserstack.appiumLogs	true
browserName	chrome

Different logs detail we can get if from here

Text Logs, Console Logs, Network Logs, Selenium Logs



**Note:** We can direct report bug in JIRA after integrating Jira. We have Report bug option at top right corner next to delete session.



BrowserStack

## APP LIVE

Interactive mobile app testing



A cartoon illustration of a man with dark hair and glasses, wearing a blue shirt and black pants. He is standing next to a large smartphone or tablet device. He is holding a small orange square object with a grid pattern on it, which appears to be a remote control or a small screen. He is pointing his right arm towards the large screen of the device, as if interacting with it. The background is a light blue gradient.



It helps to Instant access to Real Device Cloud and Test native and hybrid apps on its wide range of physical mobile and tablet devices for the most accurate testing results.

# List of Browser Stack Real Mobile & Tablet Devices for App Live Testing

## Samsung

Galaxy S21	11.0
Galaxy S21+	11.0
Galaxy S21 Ultra	11.0
Galaxy S20	10.0
Galaxy S20+	10.0
Galaxy S20 Ultra	10.0
Galaxy A51	10.0
Galaxy A11	10.0
Galaxy Note 20 Ultra	10.0
Galaxy Note 20	10.0
Galaxy S10e	9.0
Galaxy S10	9.0
Galaxy S10+	9.0
Galaxy S9+	9.0
Galaxy S8+	9.0
Galaxy A10	9.0
Galaxy Note 10	9.0
Galaxy Note 10+	9.0
Galaxy Note 9	8.1
Galaxy S9	8.0
Galaxy S9+	8.0
Galaxy J7 Prime	8.0
Galaxy A8	7.1
Galaxy Note 8	7.1
Galaxy S8	7.0
Galaxy S8+	7.0
Galaxy S7	6.0
Galaxy S6	5.0
Galaxy S5	4.4
Galaxy Tab S7	11.0
Galaxy Tab S5e	9.0
Galaxy Tab S6	9.0
Galaxy Tab S3	8.0
Galaxy Tab S4	8.0

## Google

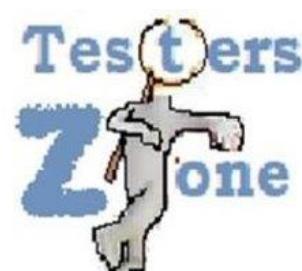
Pixel 5	12.0
Pixel 5	11.0
Pixel 4	11.0
Pixel 4	10.0
Pixel 4 XL	10.0
Pixel 3	10.0
Pixel 3a XL	9.0
Pixel 3a	9.0
Pixel 3 XL	9.0
Pixel 3	9.0
Pixel 2	9.0
Pixel 2	8.0
Pixel	8.0
Nexus 6	6.0
Nexus 6	5.0
Nexus 5	4.4
Nexus 7	6.0
Nexus 9	5.1

## Others

<b>OnePlus</b>	
9	11.0
8	10.0
7T	10.0
7	9.0
6T	9.0
<b>Xiaomi</b>	
Redmi Note 9	10.0
Redmi Note 7	9.0
Redmi Note 8	9.0
<b>Motorola</b>	
Moto G9 Play	10.0
Moto G7 Play	9.0
Moto X 2nd Gen	6.0
<b>Vivo</b>	
Y50	10.0
<b>Oppo</b>	
Reno 3 Pro	10.0
<b>Huawei</b>	
P30	9.0



Android





iPhone	
iPhone XS	15.0
iPhone 11 Pro	15.0
iPhone 8	15.0
iPhone XS	14.0
iPhone 11	14.0
iPhone 11 Pro Max	14.0
iPhone 12 Pro	14.0
iPhone 12	14.0
iPhone 12 Mini	14.0
iPhone 12 Pro Max	14.0
iPhone SE 2020	13.0
iPhone 8	13.0
iPhone XS	13.0
iPhone 11	13.0
iPhone 11 Pro	13.0
iPhone 11 Pro Max	13.0
iPhone 8 Plus	12.0
iPhone 6S	12.0
iPhone 7	12.0
iPhone 8	12.0
iPhone XS Max	12.0
iPhone XS	12.0
iPhone XR	12.0
iPhone 6	11.0
iPhone 6S	11.0
iPhone 6S Plus	11.0
iPhone SE	11.0
iPhone 8	11.0
iPhone 8 Plus	11.0
iPhone X	11.0
iPhone 7	10.0
iPhone 6S Plus	9.0
iPhone 6S	9.0
iPhone 6 Plus	8.0
iPhone 6	8.0

iPad	
iPad Pro 12.9 2018	15.0
iPad 8th	14.0
iPad Air 4	14.0
iPad Pro 11 2021	14.0
iPad Pro 12.9 2020	14.0
iPad Pro 12.9 2021	14.0
iPad 7th	13.0
iPad Air 2019	13.0
iPad Mini 2019	13.0
iPad Pro 11 2020	13.0
iPad Pro 12.9 2018	13.0
iPad Pro 12.9 2020	13.0
iPad Air 2019	12.0
iPad Mini 2019	12.0
iPad Pro 11 2018	12.0
iPad Pro 12.9 2018	12.0
iPad 6th	11.0
iPad 5th	11.0
iPad Mini 4	11.0
iPad Pro 9.7 2016	11.0
iPad Pro 12.9	11.0
iPad Air 2	8.0
iPad Mini 3	8.0



iOS

← → C [app-live.browserstack.com/#os=iOS&os\\_version=14.0&zoom\\_to\\_fit=true&full\\_screen=true&speed=1](https://app-live.browserstack.com/#os=iOS&os_version=14.0&zoom_to_fit=true&full_screen=true&speed=1) ⭐ G M :

 BrowserStack App Live Live Automate App Automate Percy More  Invite my team  Plans and pricing  Get help  1  Buy a plan

This says about how many free access time is remaining for free account.

21 mins left on your Free Trial.

Buy a plan

**SELECT SOURCE**

- Test with a Sample app
- Upload your App  
Upload your .ipa , .apk , .aab files here.
- Sync with App Center
- Install via TestFlight  
App will open directly after selecting a device
- Install via Play Store
- Install via App Store  
Available in paid plans

FREE TRIAL REAL DEVICES	ANDROID REAL DEVICES	iOS REAL DEVICES
iPhone (35)		iPhone 12 Pro 14 iPhone XR 12 iPhone 7 10
iPad (23)		iPhone 12 14 iPhone 8 15 iPhone 6S 9
		iPhone 12 Mini 14 iPhone 8 13 iPhone 6S Plus 9
		iPhone 12 Pro Max 14 iPhone 8 12 iPhone 6 8
		iPhone XS 15 iPhone 7 12 iPhone 6 Plus 8
		iPhone XS 14 iPhone 6S 12
		iPhone XS 13 iPhone X 11
		iPhone XS 12 iPhone 8 11
		iPhone XS Max 12 iPhone 8 Plus 12
		iPhone 11 Pro 15 iPhone 8 Plus 11
		iPhone 11 Pro 13 iPhone SE 2020 13
		iPhone 11 14 iPhone SE 11
		iPhone 11 13 iPhone 6S 11
		iPhone 11 Pro Max 14 iPhone 6S Plus 11
		iPhone 11 Pro Max 13 iPhone 6 11

If we don't have app or apk we can test the sample app available here

We can upload our real apk or ipa file here for testing you can get detail here: <https://www.browserstack.com/docs/app-live/integrations/microsoft-visual-studio-app-center>

We can take help of testFlight and Play store to launch the app. need to select the device and perform login first.

available only with paid plan. this is also one of the way to install the app

## App Live home screen

We can install the app from different sources for the testing as mentioned under select source section



ⓘ Upload your .ipa, .apk, .aab files here.

Uploaded Apps    Upload    URL

Show my apps only

QA.apk    Uploading file 52 / 83 MB...

FREE TRIAL REAL DEVICES    ANDROID REAL DEVICES    iOS REAL DEVICES

Category	Device	OS Version	Count
Google (22)	Pixel 2	9	Nexus 7
Samsung (40)	Pixel 5	11	
	Pixel 5	12	
OnePlus (5)	Pixel 4	11	
Motorola (5)	Pixel 4	10	
Xiaomi (3)	Pixel 4 XL	10	
	Pixel 3a XL	9	
Vivo (1)	Pixel 3a	9	
Oppo (1)	Pixel 3 XL	9	
Huawei (1)	Pixel 3	9	
	Pixel 3	10	
	Pixel 2	8	
	Pixel	8	
	Nexus 5	4.4	
	Nexus 6	6	
	Nexus 6	5	
	Nexus 9	5.1	

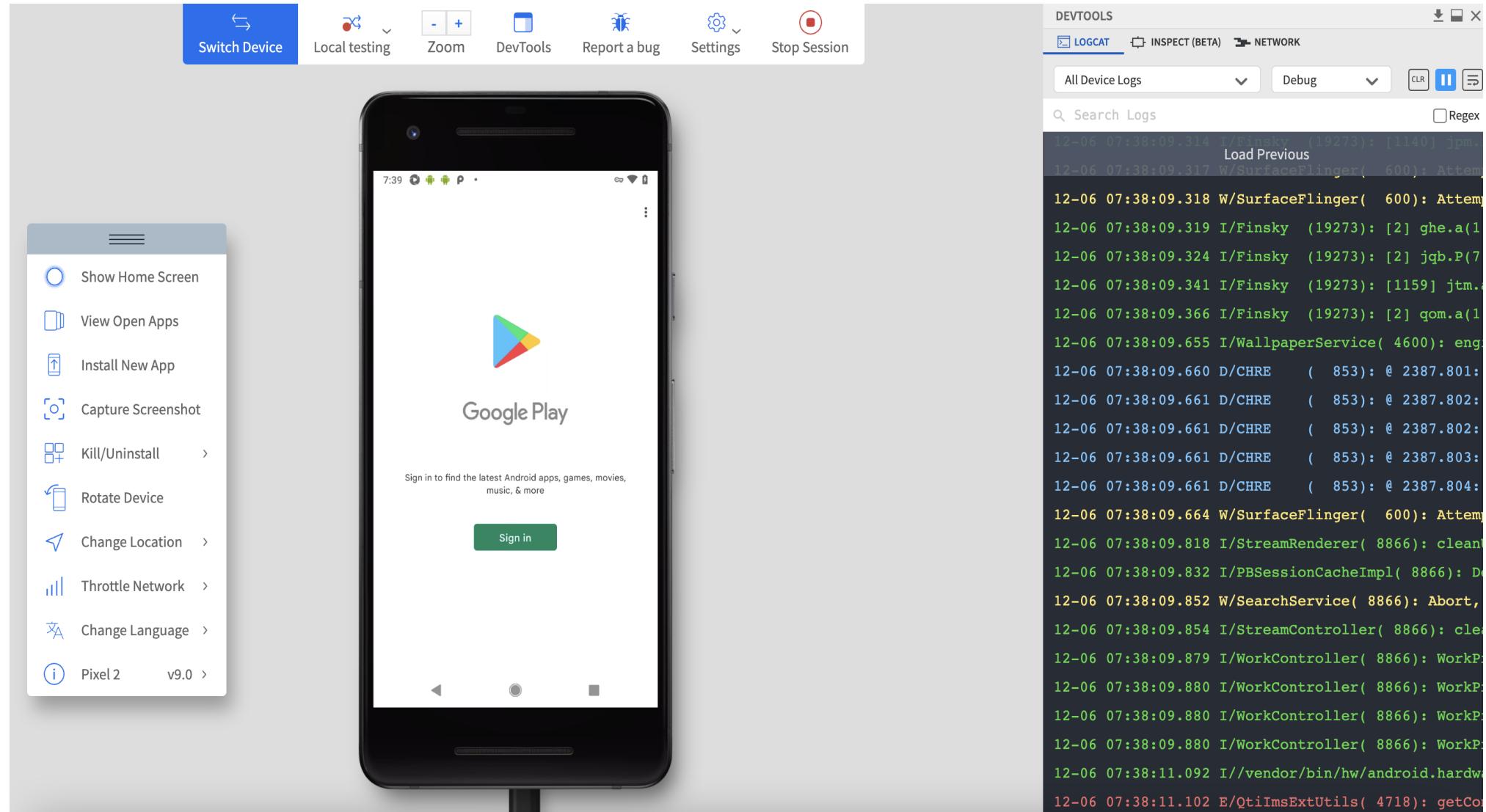
< VIEW ALL SOURCES   

## Upload your app option:

We can select the upload your app option and click on upload option it will ask for apk location and once you select it will start download. Later we need to select one device and it will launch the application in same device.

**Install via play store:** you need to click on install via play store option and select test device. Later it will ask for sign in credential and then you can install it from play store.

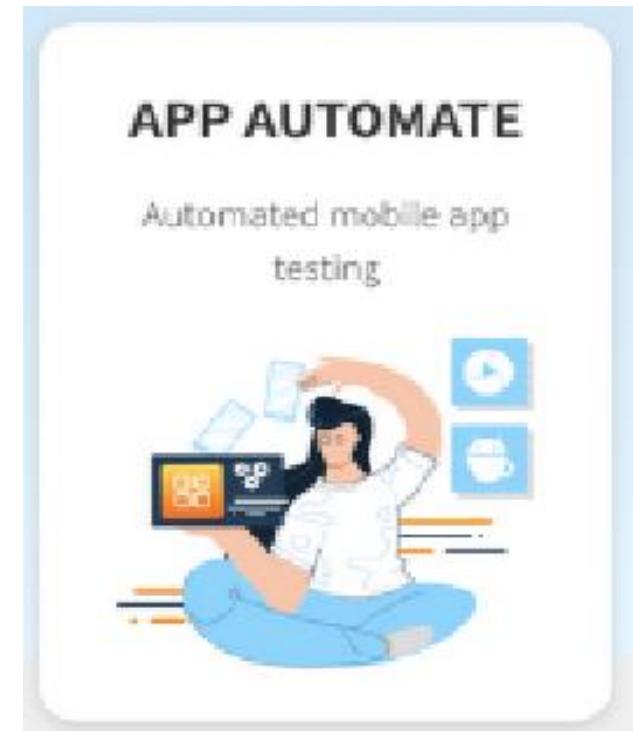
**Note:** Sign in credential is require for test flight also.





BrowserStack

- Browser Stack App Automate enables you to test native and hybrid mobile applications using **Appium automation framework**. It's easy to run your Appium tests written in Java on real Android and iOS devices on Browser Stack.



## Run your first Build



Run a sample TestNG test build on Browser Stack by changing a few lines in the code using these simple steps:

### STEP 1

#### Setup your environment

Ensure that you have installed Appium's Java client library. Follow detailed instructions here if you are using Gradle or Maven to build your Java project.

- i You need to have `maven` and Java installed in your system to be able to run our sample build.

```
// Maven users can add this dependency to project's POM
<dependency>
<groupId>io.appium</groupId>
<artifactId>java-client</artifactId>
<version>7.0.0</version>
</dependency>
```



# Step 2:

## Upload Your app

Use one of the following options to upload an Android app (`.apk` or `.aab` file) or iOS app (`.ipa` file) to BrowserStack servers. The generated `app_url` is a unique ID used to identify the uploaded app build. Take note of this `app_url` as it will be used in steps ahead.

[Use Sample Apps](#)

[Upload via file manager](#)

[Upload via REST API](#)

If you do not have an Android app or an iOS app at present, use one of our sample apps to run your first Appium test on BrowserStack App Automate. Choose from the options below

 SampleAndroidApp.apk

- OR -

 SampleiOSApp.ipa

### Example:

```
/* Note the "app_url" value for the sample app. This value uniquely identifies the app on Browser Stack. */
```

```
{"app_url":"bs://c700ce60cf13ae8ed97705a55b8e022f13c5827c"}
```

```
/* In your test script, use this "app_url" value to specify the application under test using the "app" capability. During test execution, the sample app will automatically be installed and launched on the device being tested. */
```

```
caps.setCapability("app",  
"bs://c700ce60cf13ae8ed97705a55b8e022f13c5827c")
```

## Steps 3:

### Configure test script

- **Using desired capabilities**

Desired capabilities are a series of key-value pairs that allow you to configure your Appium tests on Browser Stack. Let's start with most important ones.

- **Set access credentials**

Use browser stack. user and browser stack. key variables to set your Browser Stack access credentials This is required to authenticate your tests.

- **Specify application under test**

Use the app capability to specify your uploaded app that will be installed on device during test execution. Set its value using the app\_url obtained in Step 2.

- **Set the device to run the test on**

Use the device and os\_version capability to specify the device for testing.

- a. Android. b. Google Pixel 3[observe different available devices in next couple of slides]

- **Create remote web driver**

Next, initialize a remote web driver to start a new test session on Browser Stack. You are all set to remotely test your app on Browser Stack devices!

- **Write a test case**

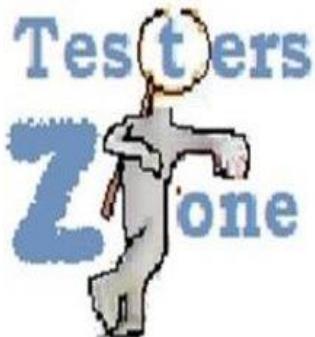
Add an existing test case or write a new one using different Appium commands. We have provided a test case for our sample app but make sure you edit it to test your own app !





Android

## List of Browser Stack Real Mobile & Tablet Devices for App Automate Testing



Samsung	Google	Others
Samsung Galaxy S21 Ultra 11.0	Google Pixel 6 12.0	OnePlus
Samsung Galaxy S21 11.0	Google Pixel 5 12.0	OnePlus 9 11.0
Samsung Galaxy S21 Plus 11.0	Google Pixel 5 11.0	OnePlus 8 10.0
Samsung Galaxy S20 10.0	Google Pixel 4 11.0	OnePlus 7T 10.0
Samsung Galaxy S20 Plus 10.0	Google Pixel 4 XL 10.0	OnePlus 7 9.0
Samsung Galaxy S20 Ultra 10.0	Google Pixel 4 10.0	OnePlus 6T 9.0
Samsung Galaxy Note 20 Ultra	Google Pixel 3 10.0	
Samsung Galaxy Note 20 10.0	Google Pixel 3a XL 9.0	Xiaomi
Samsung Galaxy A51 10.0	Google Pixel 3a 9.0	Xiaomi Redmi Note 9 10.0
Samsung Galaxy A11 10.0	Google Pixel 3 XL 9.0	Xiaomi Redmi Note 8 9.0
Samsung Galaxy S9 Plus 9.0	Google Pixel 3 9.0	Xiaomi Redmi Note 7 9.0
Samsung Galaxy S8 Plus 9.0	Google Pixel 2 9.0	
Samsung Galaxy S10e 9.0	Google Pixel 2 8.0	Motorola
Samsung Galaxy S10 Plus 9.0	Google Pixel 1 8.0	Motorola Moto G9 Play 10.0
Samsung Galaxy S10 9.0	Google Pixel 1 7.1	Motorola Moto G7 Play 9.0
Samsung Galaxy Note 10 Plus 9.0	Google Nexus 6 6.0	Motorola Moto X 2nd Gen 6.0
Samsung Galaxy Note 10 9.0	Google Nexus 6 5.0	
Samsung Galaxy Note 10 9.0	Google Nexus 5 4.4	Vivo
Samsung Galaxy A10 9.0		Y50 10.0
Samsung Galaxy Note 9 8.1		Oppo
Samsung Galaxy J7 Prime 8.1		Reno 3 Pro 10.0
Samsung Galaxy S9 Plus 8.0		Huawei
Samsung Galaxy S9 8.0		P30 9.0
Samsung Galaxy Note 8 7.1		
Samsung Galaxy A8 7.1		
Samsung Galaxy S8 Plus 7.0		
Samsung Galaxy S8 7.0		
Samsung Galaxy S7 6.0		
Samsung Galaxy S6 5.0		
Samsung Galaxy S5 4.4		
Samsung Galaxy Tab S7 10.0	Tablet	
Samsung Galaxy Tab S6 9.0	Tablet	
Samsung Galaxy Tab S5e 9.0	Tablet	
Samsung Galaxy Tab S4 8.1	Tablet	
Samsung Galaxy Tab S3 8.0	Tablet	
Samsung Galaxy Tab 4 4.4	Tablet	

Physical Mobile Device

iPhone	
iPhone XS	15
iPhone 13	15
iPhone 11 Pro	15
iPhone XS	14
iPhone 12 Pro Max	14
iPhone 12 Pro	14
iPhone 12 Mini	14
iPhone 12	14
iPhone 11 Pro Max	14
iPhone 11	14
iPhone XS	13
iPhone 11 Pro Max	13
iPhone 11 Pro	13
iPhone 11	13
iPhone XS	12
iPhone XS Max	12
iPhone XR	12
iPhone X	11
iPhone 8	15
iPhone 8	13
iPhone 8	12
iPhone 8	11
iPhone 8 Plus	12
iPhone 8 Plus	11
iPhone 7	12
iPhone 7	10
iPhone 7 Plus	10
iPhone 6S	12
iPhone 6S	11
iPhone 6S Plus	11
iPhone 6	11
iPhone SE 2020	13
iPhone SE	11

iPad	
iPad Air 4	14
iPad Pro 12.9 2021	14
iPad Pro 12.9 2020	14
iPad Pro 11 2021	14
iPad Pro 12.9 2020	13
iPad Pro 12.9 2018	15
iPad Mini 2021	15
iPad 8th	14
iPad Pro 12.9 2018	13
iPad Pro 11 2020	13
iPad Mini 2019	13
iPad Air 2019	13
iPad 7th	13
iPad Pro 12.9 2018	12
iPad Pro 11 2018	12
iPad Mini 2019	12
iPad Air 2019	12
iPad Pro 9.7 2016	11
iPad Pro 12.9 2017	11
iPad Mini 4	11
iPad 6th	11
iPad 5th	11

```

package android;
import java.net.URL;
import java.util.List;
import java.util.function.Function;
import java.net.MalformedURLException;
import io.appium.java_client.MobileBy;
import io.appium.java_client.android.AndroidDriver;
import io.appium.java_client.android.AndroidElement;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
public class BrowserStackSample {

    public static void main(String[] args) throws MalformedURLException, InterruptedException {
        DesiredCapabilities caps = new DesiredCapabilities();
        // Set your access credentials
        caps.setCapability("browserstack.user", "mithileshsingh_D98GPT");
        caps.setCapability("browserstack.key", "DvpabBSodXe2okIk4CCd");
        // Set URL of the application under test
        caps.setCapability("app", "bs://c700ce60cf13ae8ed97705a55b8e022f13c5827c");
        // Specify device and os_version for testing
        caps.setCapability("device", "Google Pixel 3");
        caps.setCapability("os_version", "9.0");
        // Set other BrowserStack capabilities
        caps.setCapability("project", "First Java Project");
        caps.setCapability("build", "browserstack-build-1");
        caps.setCapability("name", "first_test");
    }
}

```



```

// Initialise the remote Webdriver using BrowserStack remote URL
// and desired capabilities defined above
AndroidDriver<AndroidElement> driver = new AndroidDriver<AndroidElement>(
    new URL("http://hub.browserstack.com/wd/hub"), caps);

// Test case for the BrowserStack sample Android app.
// If you have uploaded your app, update the test case here.
AndroidElement searchElement = (AndroidElement) new WebDriverWait(driver,
30).until(
    ExpectedConditions.elementToBeClickable(
        MobileBy.AccessibilityId("Search Wikipedia")));
searchElement.click();
AndroidElement insertTextElement = (AndroidElement)
new WebDriverWait(driver, 30).until(
    ExpectedConditions.elementToBeClickable(
        MobileBy.id("org.wikipedia.alpha:id/search_src_text")));
insertTextElement.sendKeys("BrowserStack");
Thread.sleep(5000);
List<AndroidElement> allProductsName = driver.findElementsByClassName(
    "android.widget.TextView");
assert(allProductsName.size() > 0);

// Invoke driver.quit() after the test is done to indicate that the test is
completed.
driver.quit();
}

```

#### Example code

this is applicable in my case, Browser stack will suggest code for you once you will login and navigate to app automate option and select your OS and device.

## Step 4:

### Execute your test script



- You are now ready to run your first Appium test on Browser Stack. Copy the code snippet on right side and paste it in your local file. On your local machine, open the terminal/command prompt and navigate to the folder containing your test script. Build and run the test script just like any other Java program using your project's preferred build tools (E.g. Maven, Gradle).

```
# Using Maven  
mvn test -P <android-first-test>
```

# View test results on dashboard



Session Listing    Session Details

The screenshot shows the Browser Stack App Automate dashboard. On the left, there's a sidebar with a 'FILTERS' dropdown and three session cards:

- AA sample script**: Status green, 2 SESSIONS, 20 mins ago.
- Build #8760: BStackMediaApp.ipa v1.0**: Status green, 1/1 PASSED, 30 mins ago.
- QA/AppAutomateProductionSanity appium\_ios 7680**: Status red, 4/17 FAILED, 38 mins ago.

A blue callout box with a circular arrow icon and the number '1' is pointing to the 'FILTERS' dropdown. The callout contains the text: "Access all builds and filter them using user, project, status, team, and framework." A 'Next →' button is visible at the bottom right of the callout.

In the main area, there are two rows of session cards. The first row includes a card for "Build #8760: BStackMediaApp.ipa v1.0" and another for "QA/AppAutomateProductionSanity appium\_ios 7680". The second row is partially visible. Circular icons with numbers '2' and '3' are placed above the second and third cards respectively in the first row.

**Text at the bottom:**

This is same which we discussed in Browser Stack automate option. That was for web testing and this (app automate) is for mobile testing

# Browser Stack Integrations

Cypress.io

Jenkins

Selenium

Bitbucket

Circle CI

GitHub

Travis CI

WordPress

Visual Studio IDE

Appium

Jira Software

Trello

Slack

Earl Grey

Espresso

Java script Testing

## Integrate effortlessly with testing frameworks

Integrate with BrowserStack using Appium, Espresso, XCUITest or EarlGrey.



[Appium](#)



[Espresso](#)



[XCUITest](#)



[EarlGrey](#)



# Browser Stack Competitors

- Sauce Labs
- Perfecto, by Perforce
- Experitest
- Cross Browser Testing
- Lambda Test
- AWS Device Farm

# Note:

- I am not covering Percy tool in this ppt because it's a separate topic which deals with visual testing. So will cover this in visual testing introduction ppt.
- If you want to see the end to end test framework integrated with Browser stack platform visit this link: [https://github.com/mithilesh777/softwaretesting/tree/master/Selenium\\_BS\\_integration](https://github.com/mithilesh777/softwaretesting/tree/master/Selenium_BS_integration)





Thank you

Mithilesh Singh