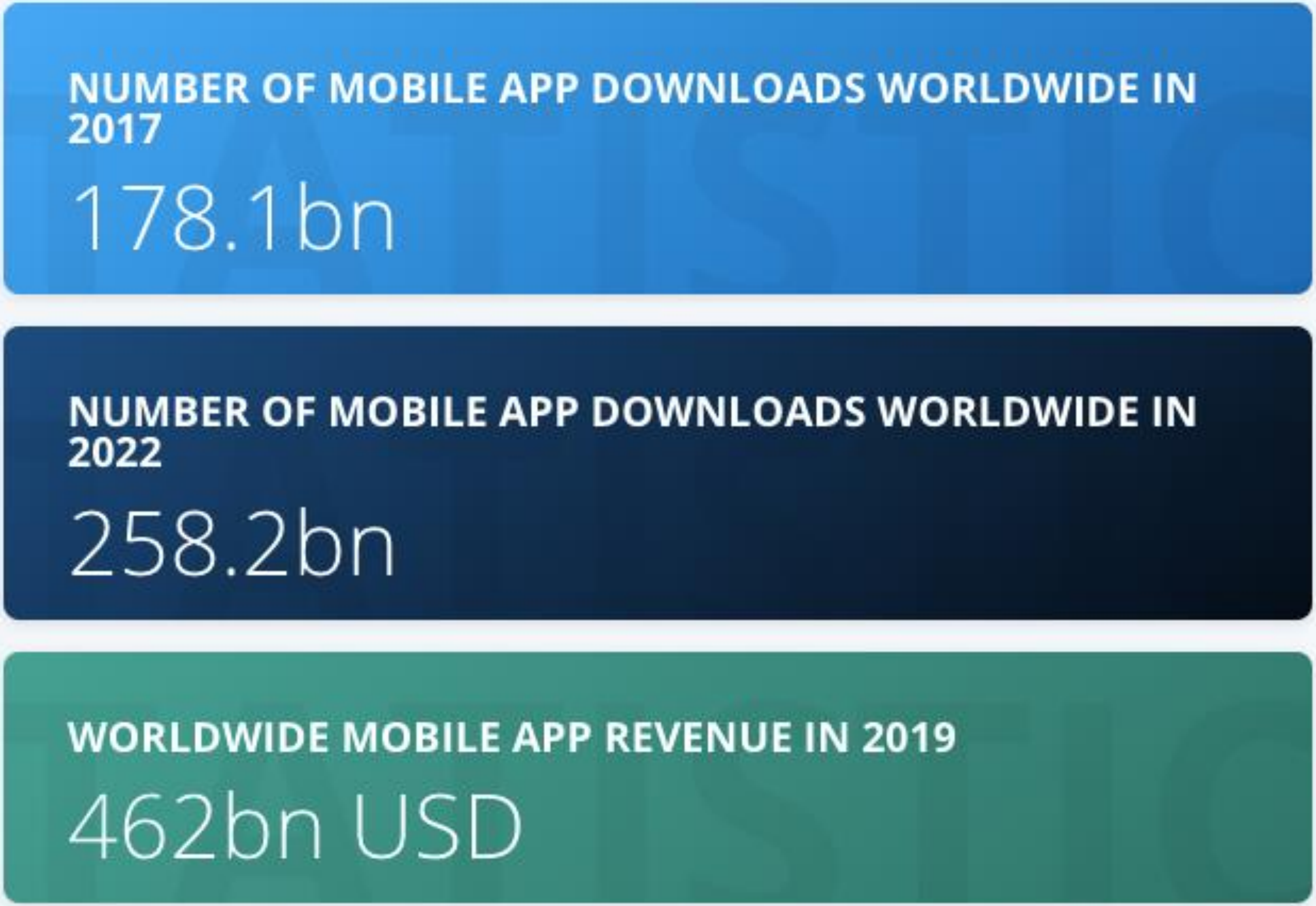
Software Testing

#08 – Mobile App Testing

Herman Kabetta

hermanka.github.io



NUMBER OF MOBILE APP DOWNLOADS WORLDWIDE IN 2017

178.1bn

NUMBER OF MOBILE APP DOWNLOADS WORLDWIDE IN 2022

258.2bn

WORLDWIDE MOBILE APP REVENUE IN 2019

462bn USD

RapidValue™
Enabling Mobility

MOBILITY TESTING

Dark Horse of the App Development World

Think about the mobile apps you use, if an app you downloaded crashed your phone, what would you do? Nearly 50% of consumers delete the app from their phone, if they find a bug.

Source: Smartbear

PREDICTIONS BY INDUSTRY EXPERTS

Through 2015, more than **75%** of mobile applications will fail basic security tests

The revenues from mobile application testing will exceed **$800** million by the end of 2017

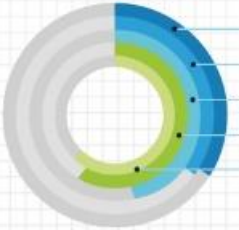
$1.5 billion net worth of the testing tool market

1.6% annual growth for software testing services industry

52% users say that a bad mobile app experience made them less likely to engage with it further

Source: Gartner, Ibisworld, Experience Dynamics

TOP REASONS TO OPTIMIZE TESTING ACTIVITIES



- 34% - Reducing costs of testing activities
- 35% - Increasing objectivity of testing activities
- 46% - Increasing testing automation
- 58% - Increasing quality and transparency of testing process
- 61% - Increasing product quality

n=279

Source: Pierre Audoin Consultants

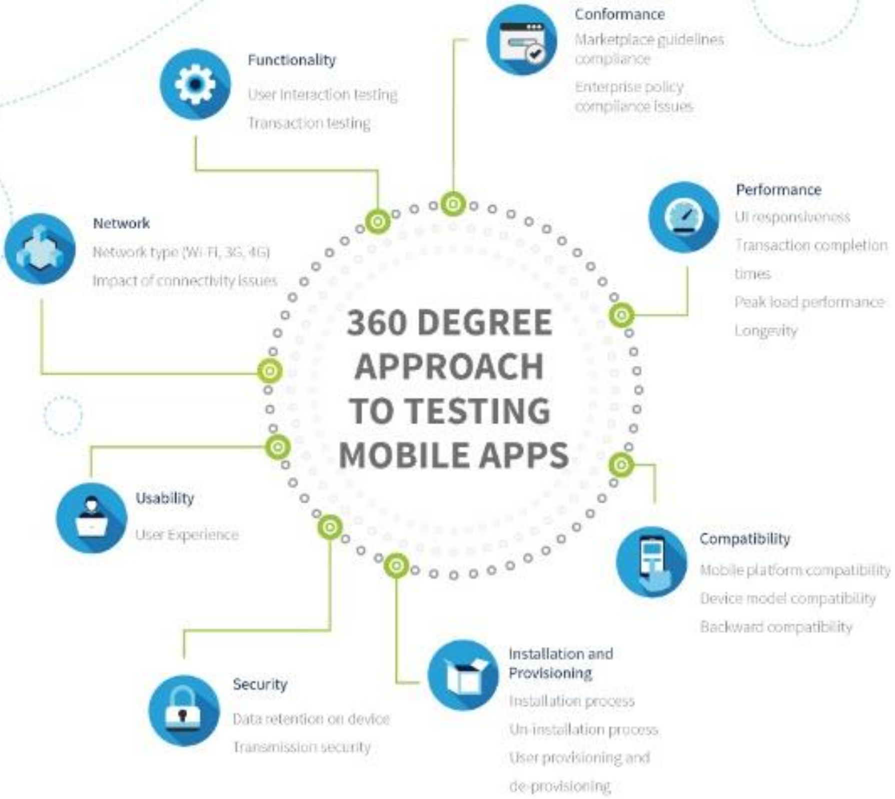
TIPS TO ENSURE SUCCESSFUL CROSS-PLATFORM TESTING

Test on Real & Multiple Devices

Use Crowdsourced Testing

Accept Testing is a Continous Process

PARAMETERS AND TYPES OF TESTING



360 DEGREE APPROACH TO TESTING MOBILE APPS

Functionality
User interaction testing
Transaction testing

Conformance
Marketplace guidelines compliance
Enterprise policy compliance issues

Network
Network type (Wi-Fi, 3G, 4G)
Impact of connectivity issues

Performance
UI responsiveness
Transaction completion times
Peak load performance
Longevity

Usability
User Experience

Compatibility
Mobile platform compatibility
Device model compatibility
Backward compatibility

Security
Data retention on device
Transmission security

Installation and Provisioning
Installation process
Un-installation process
User provisioning and de-provisioning

Mobile App Testing Tools

selendroid
SELENIUM FOR ANDROID

headspin

Kobiton

froglogic

TestProject

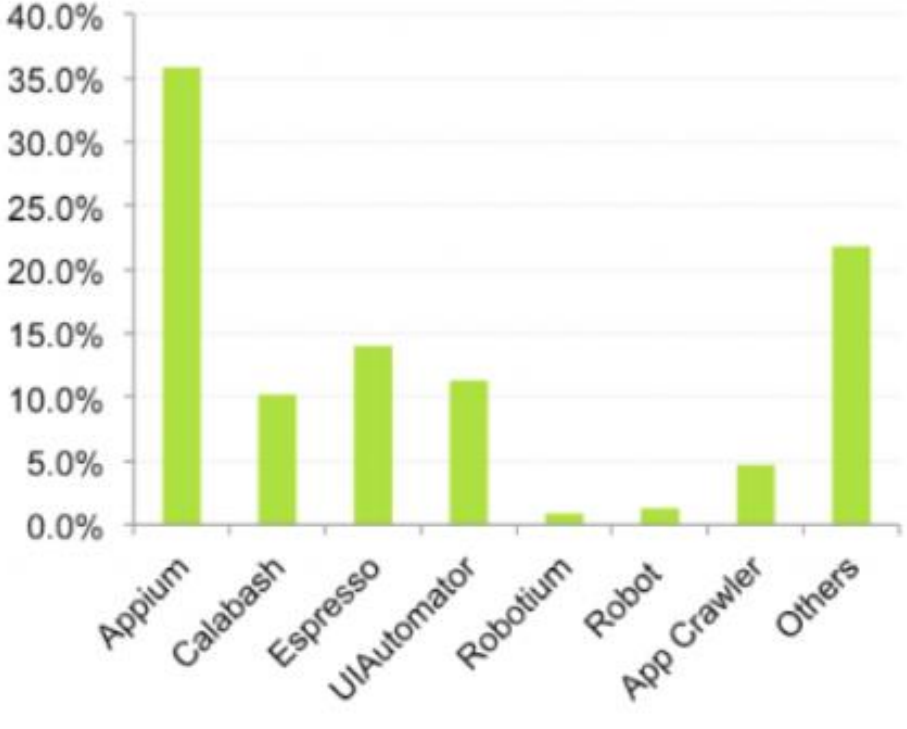
experitest

Robotium

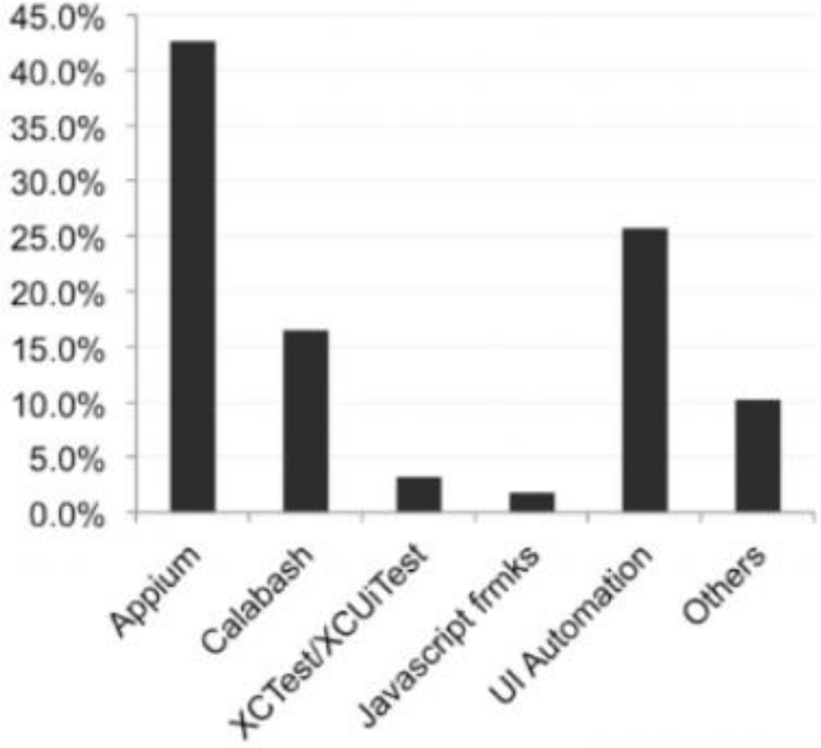
TestingBot

appium

Android

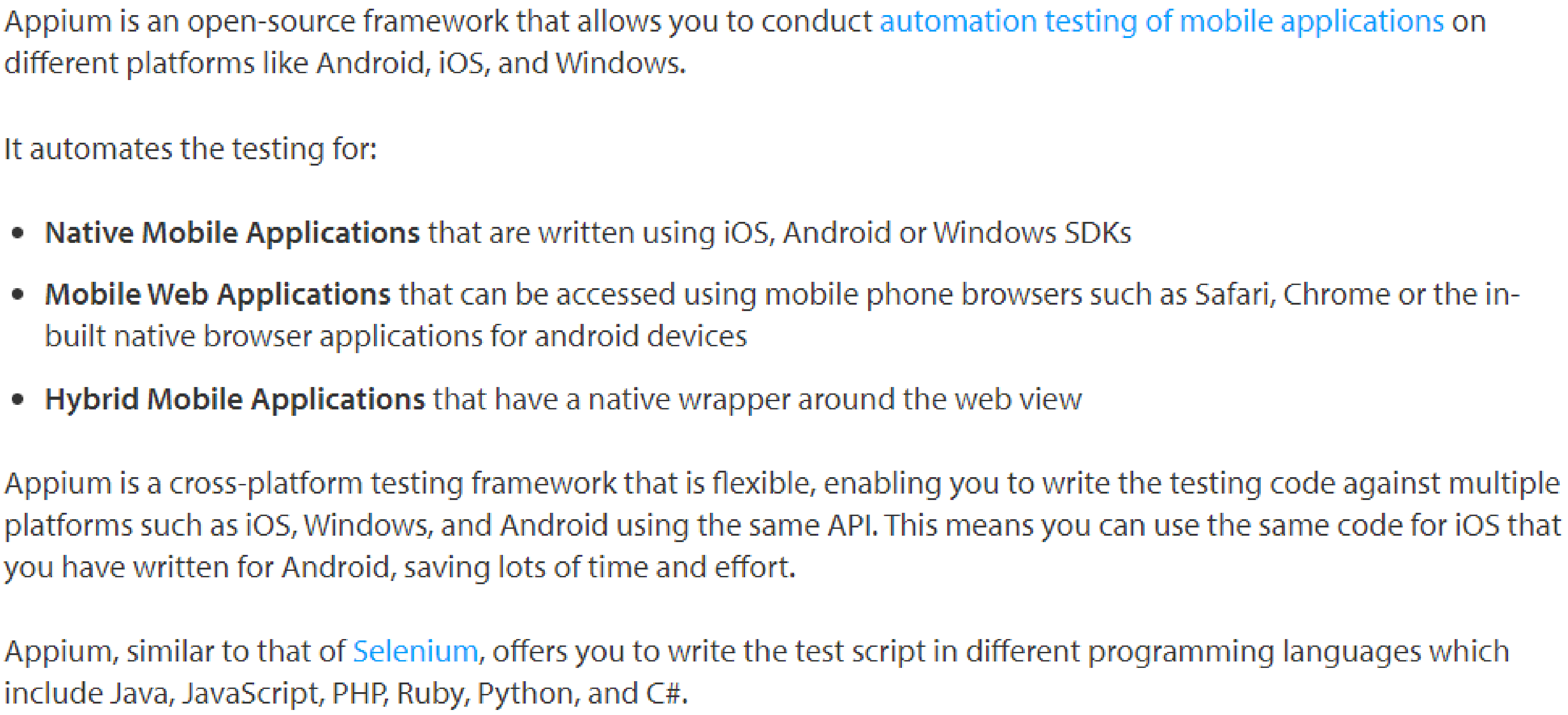


iOS



Source: Bitbar

What is Appium?



Appium is an open-source framework that allows you to conduct automation testing of mobile applications on different platforms like Android, iOS, and Windows.

It automates the testing for:

- **Native Mobile Applications** that are written using iOS, Android or Windows SDKs
- **Mobile Web Applications** that can be accessed using mobile phone browsers such as Safari, Chrome or the in-built native browser applications for android devices
- **Hybrid Mobile Applications** that have a native wrapper around the web view

Appium is a cross-platform testing framework that is flexible, enabling you to write the testing code against multiple platforms such as iOS, Windows, and Android using the same API. This means you can use the same code for iOS that you have written for Android, saving lots of time and effort.

Appium, similar to that of Selenium, offers you to write the test script in different programming languages which include Java, JavaScript, PHP, Ruby, Python, and C#.

Appium vs Selenium

#1. General

Appium

Appium is one of the popular open source for automation testing specially in any kind of native application, Mobile App like android app and IOS app, also it can be supported in hybrid application as well, where application run in both the environment android and IOS. This kind of automation testing ensures of avoiding any kind of systemic break due to any special patch installation after app working as live in production. Ensuring fill integration testing and avoid one of the huge manual effort done by the tester which done earlier based on application common test cases designed by the tester. Every time they need to manually testing each and every features of entire application, but by using Appium user can easily able to avoid the same and mainly concentrate on testing of changing impacted area for better and quick delivery with proper impact analysis anytime by this kind of automation tool

Selenium

Selenium is also one of the very popular automate testing tool mainly designed for any kind of web application , it also supported to all the current popular browsers and every operating system like Windows, LinuX or MACOS etc.

#2. Design

Appium

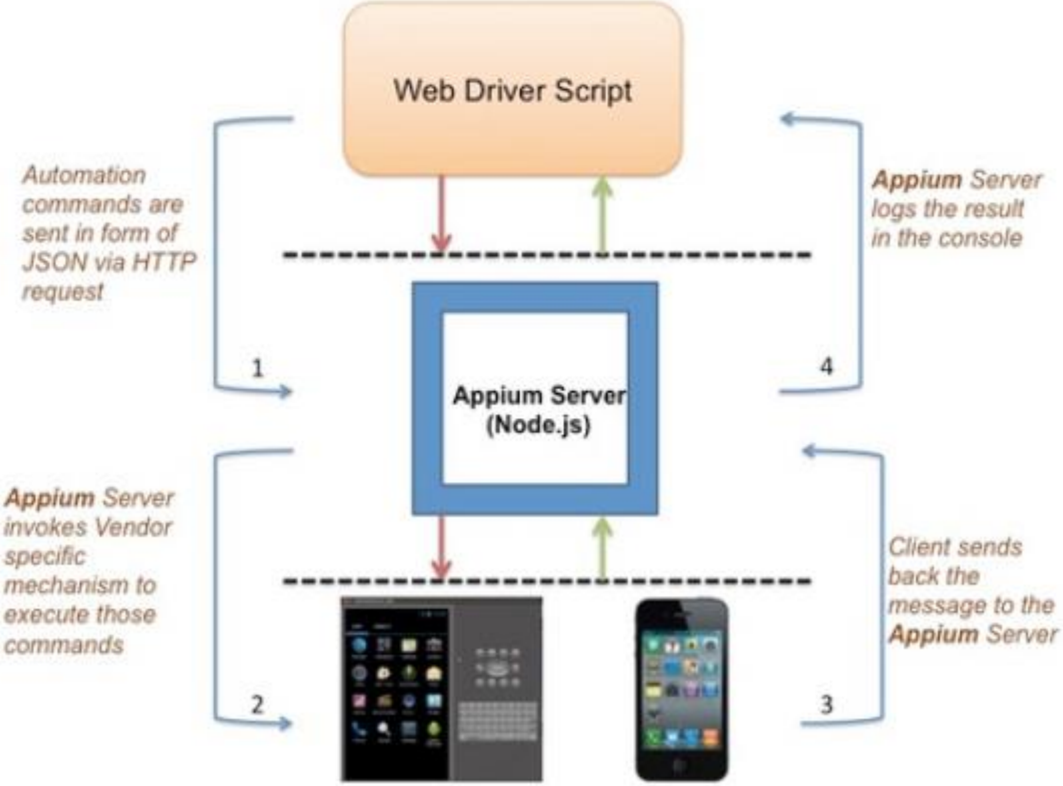
Appium is mainly designed as HTTP server, as it is going to support any kind of Mobile App, so developed as HTTP server is required for the same. But it mainly following or developing the same in node JS, not using normal Java code or JS code. So for the same reason developer who are willing to use appium for their automate testing in any kind of mobile app, they have to install Node JS in their system before starting this Appium tool. It is one kind of prerequisite for initializing Appium in the system.

Selenium

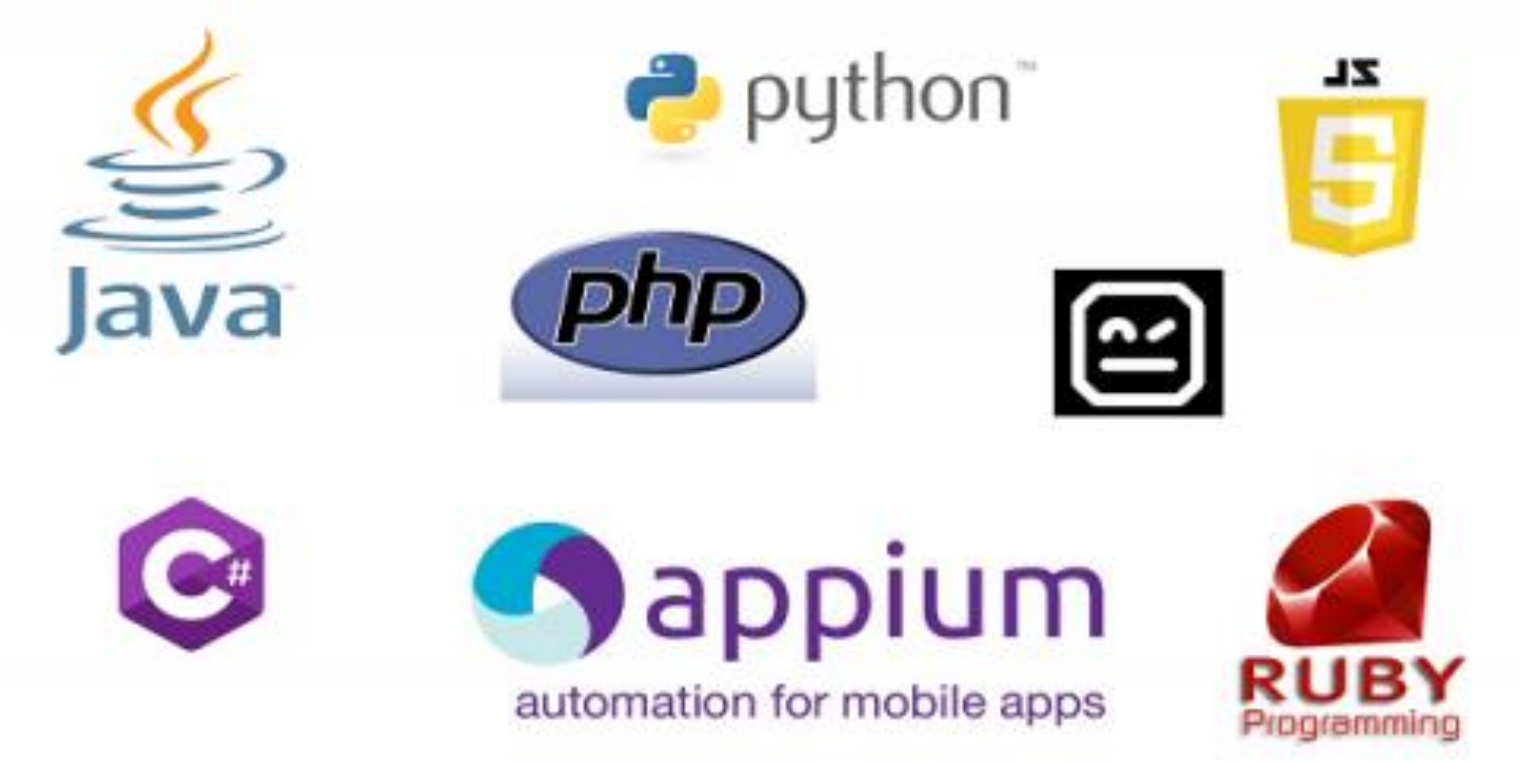
Selenium is mainly designed for maintain automation logic on specifically of web application or websites. It basically designed in that way so that it can able to interact easily with the application by using browser actions automatically.

www.educba.com

Architecture


Web Driver Script
Automation commands are sent in form of JSON via HTTP request
1
Appium Server (Node.js)
Appium Server logs the result in the console
4
Appium Server invokes Vendor specific mechanism to execute those commands
2
Client sends back the message to the Appium Server
3

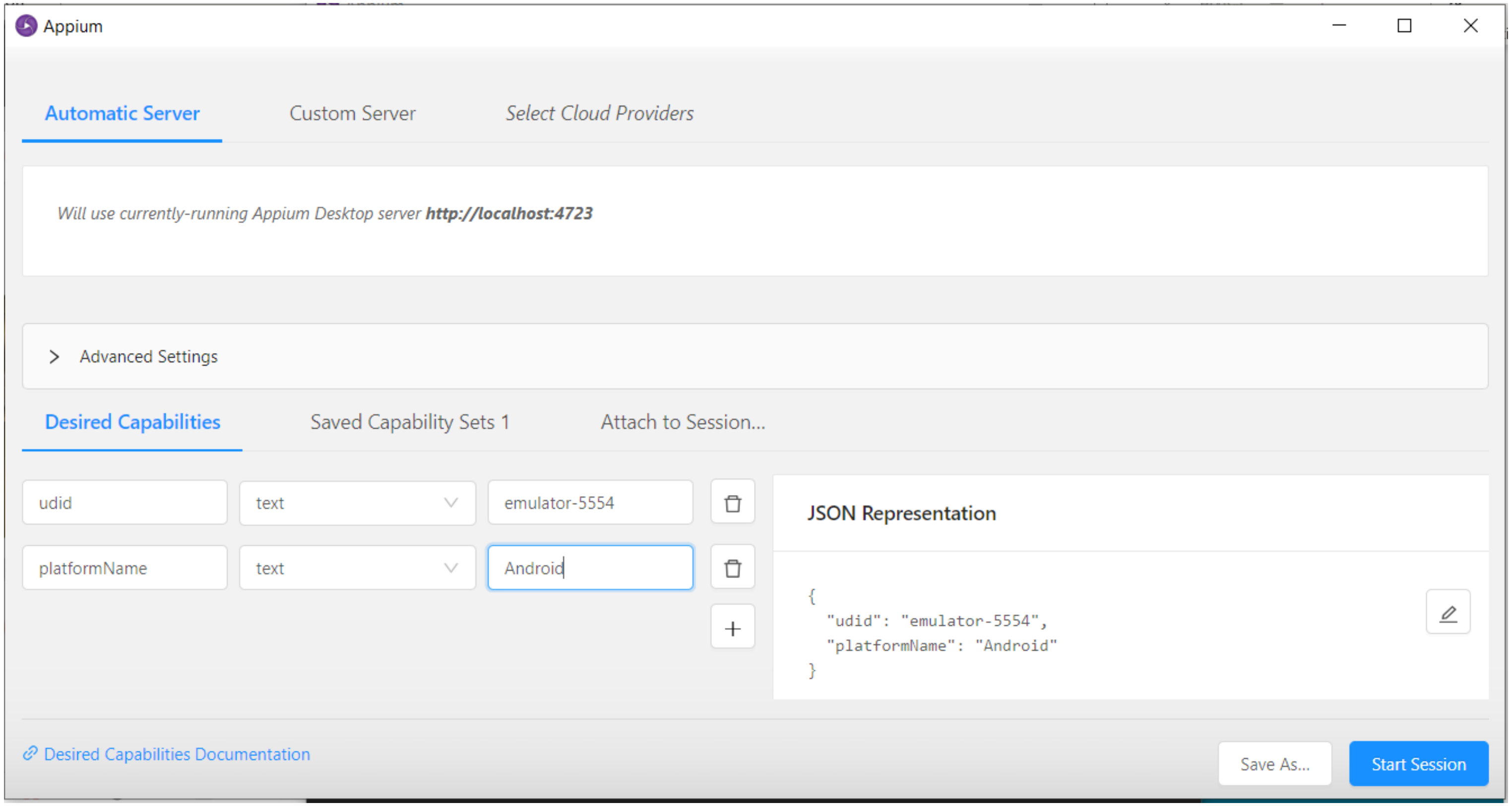
Languages Support



Installation

1. Download and Install latest version of Java 8 (JDK)
2. Setup Java(JAVA_HOME and PATH) Environment Variables
3. Download and Install Android Studio
4. Install additional Android SDK tools
5. Setup Android Environment Variables (ANDROID_HOME & ANDROID_SDK_ROOT) to android sdk
6. Download and Install Appium Desktop App

Inspect Element



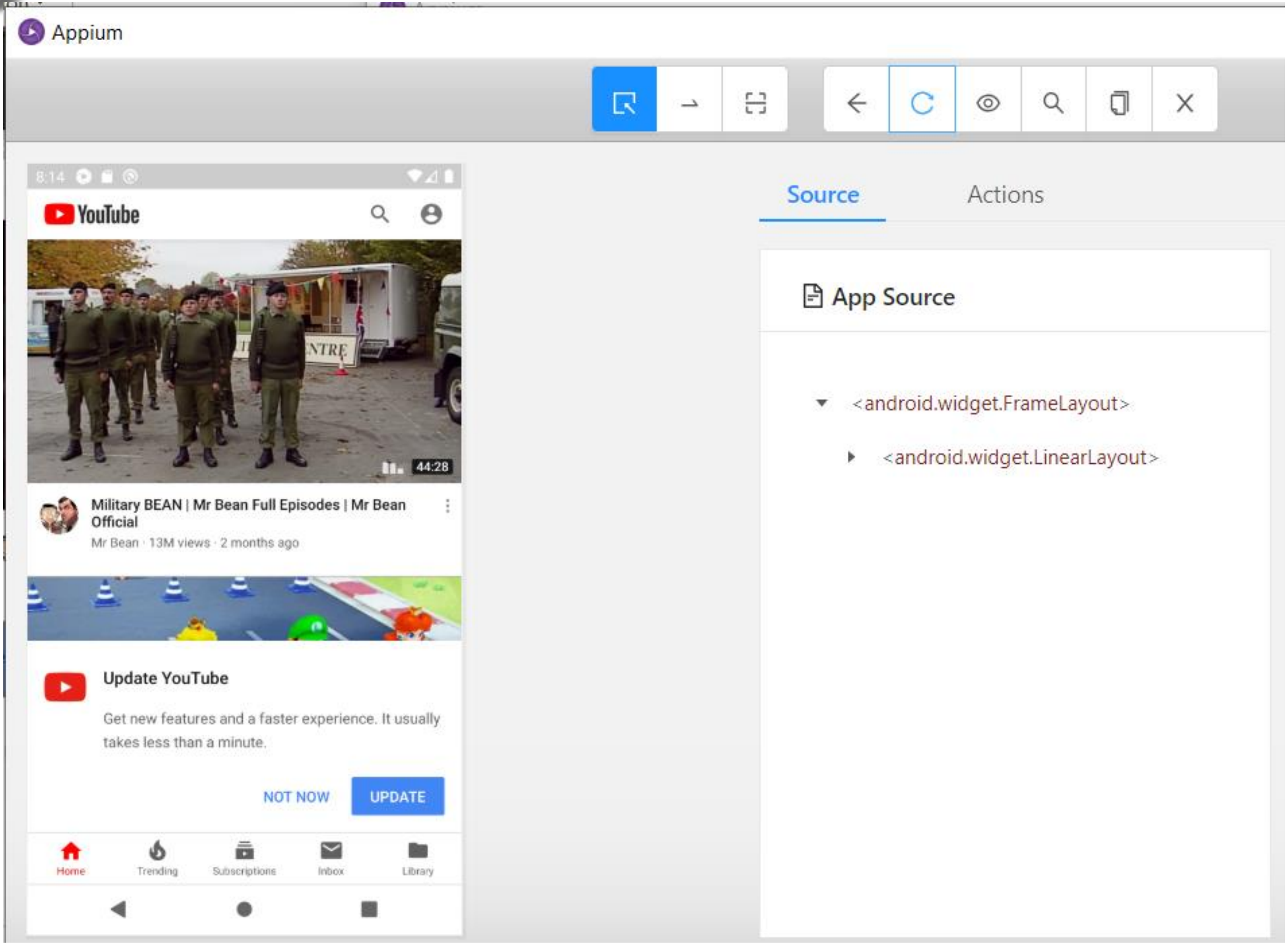
Will use currently-running Appium Desktop server ***http://localhost:4723***

udid	text	emulator-5554
platformName	text	Android

JSON Representation

```
{
  "udid": "emulator-5554",
  "platformName": "Android"
}
```

Inspect Element



Appium

Source | Actions

App Source

```
▾ <android.widget.FrameLayout>
    ▸ <android.widget.LinearLayout>
```

Military BEAN | Mr Bean Full Episodes | Mr Bean Official

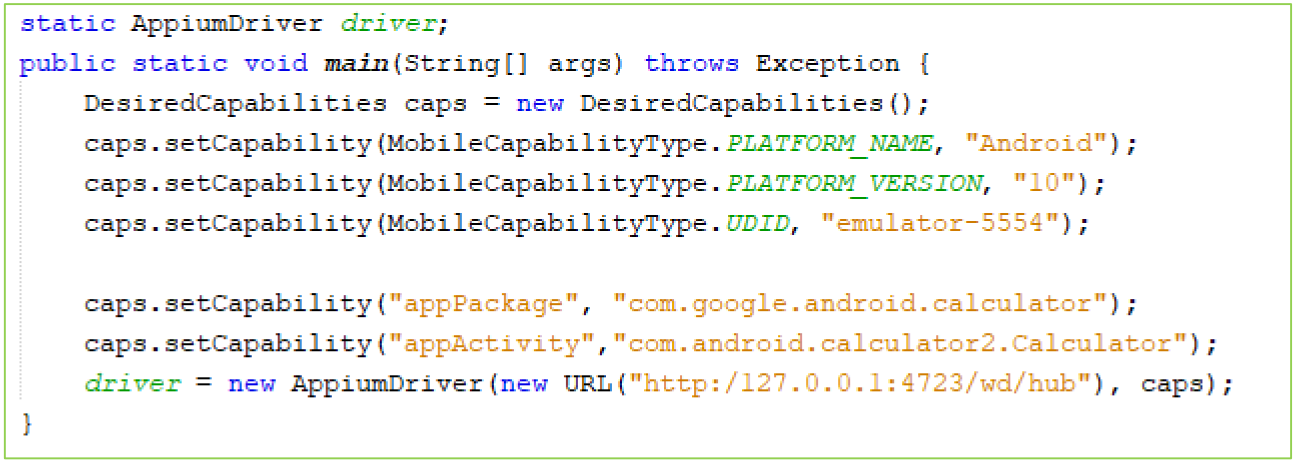
Mr Bean · 13M views · 2 months ago

Update YouTube

Get new features and a faster experience. It usually takes less than a minute.

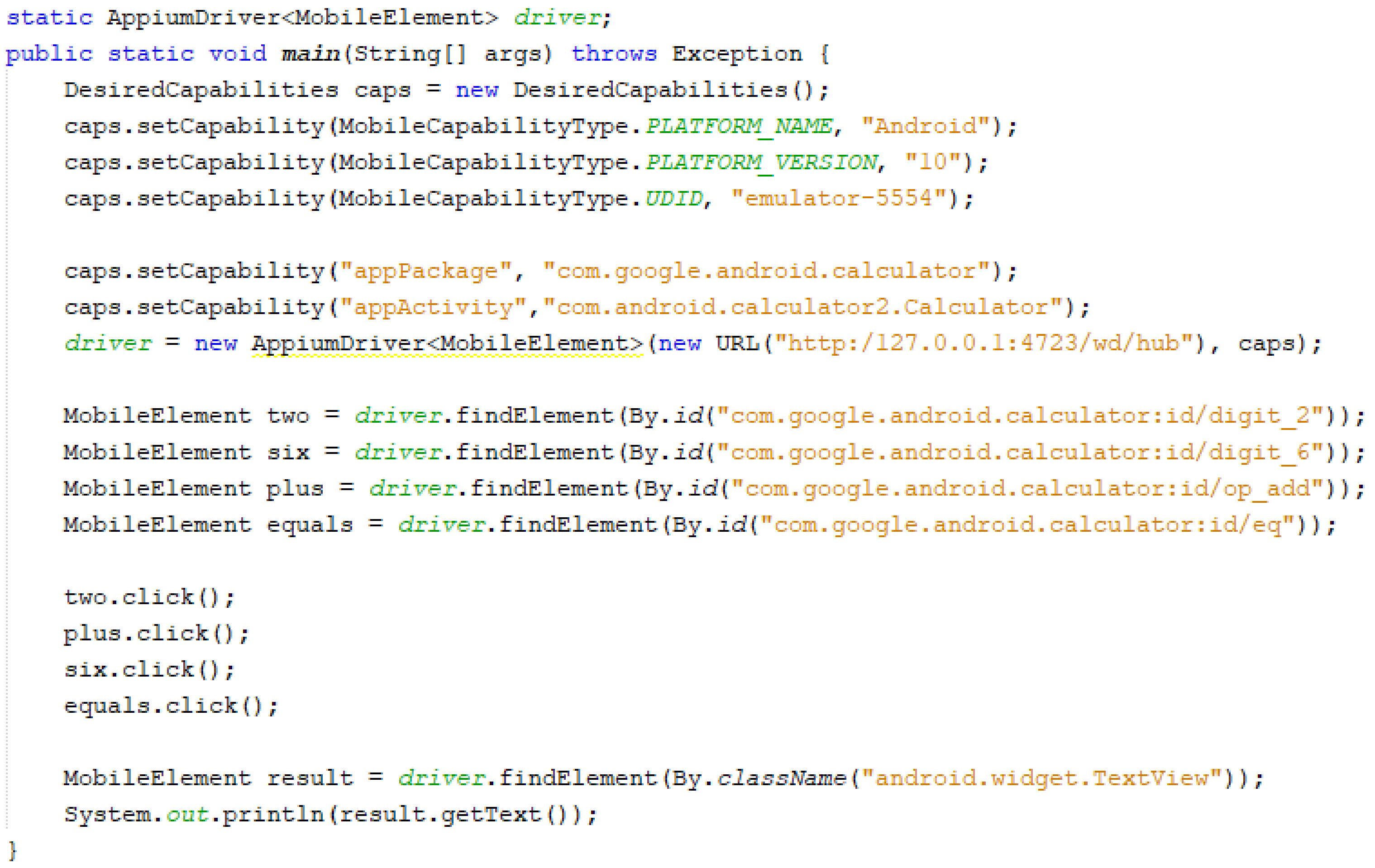
Get Started

1. Create Maven Project
2. Add Libraries
   - Selenium Java
   - Appium Java Client
   - TestNG
3. Connect device
4. Start Appium Server / Desktop
5. Code to start app on device



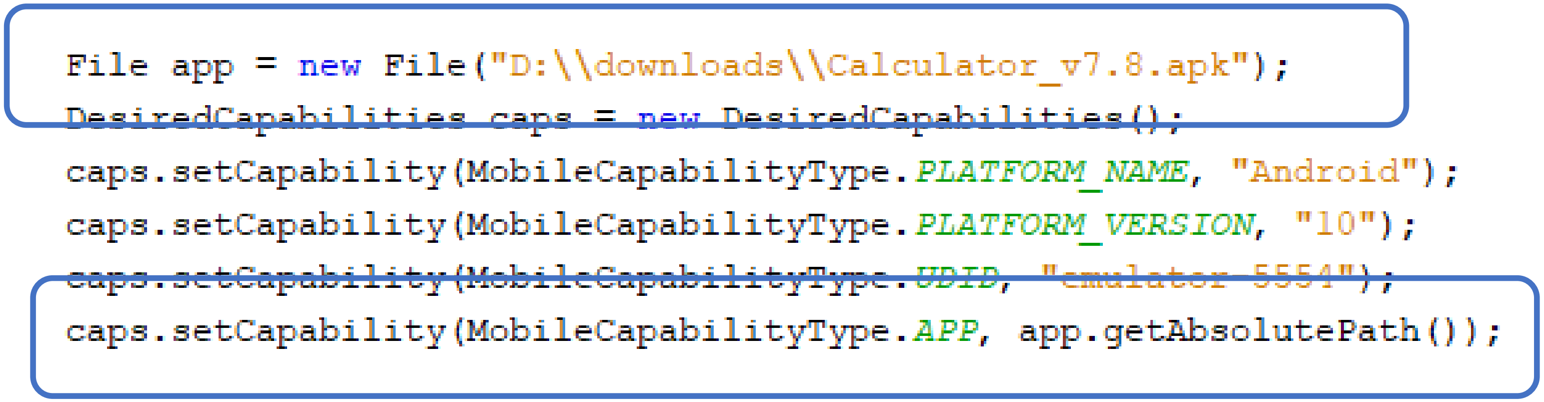
```java
static AppiumDriver driver;
public static void main(String[] args) throws Exception {
    DesiredCapabilities caps = new DesiredCapabilities();
    caps.setCapability(MobileCapabilityType.PLATFORM_NAME, "Android");
    caps.setCapability(MobileCapabilityType.PLATFORM_VERSION, "10");
    caps.setCapability(MobileCapabilityType.UDID, "emulator-5554");

    caps.setCapability("appPackage", "com.google.android.calculator");
    caps.setCapability("appActivity","com.android.calculator2.Calculator");
    driver = new AppiumDriver(new URL("http:/127.0.0.1:4723/wd/hub"), caps);
}
```

FindElement



```java
static AppiumDriver<MobileElement> driver;
public static void main(String[] args) throws Exception {
    DesiredCapabilities caps = new DesiredCapabilities();
    caps.setCapability(MobileCapabilityType.PLATFORM_NAME, "Android");
    caps.setCapability(MobileCapabilityType.PLATFORM_VERSION, "10");
    caps.setCapability(MobileCapabilityType.UDID, "emulator-5554");

    caps.setCapability("appPackage", "com.google.android.calculator");
    caps.setCapability("appActivity","com.android.calculator2.Calculator");
    driver = new AppiumDriver<MobileElement>(new URL("http:/127.0.0.1:4723/wd/hub"), caps);

    MobileElement two = driver.findElement(By.id("com.google.android.calculator:id/digit_2"));
    MobileElement six = driver.findElement(By.id("com.google.android.calculator:id/digit_6"));
    MobileElement plus = driver.findElement(By.id("com.google.android.calculator:id/op_add"));
    MobileElement equals = driver.findElement(By.id("com.google.android.calculator:id/eq"));

    two.click();
    plus.click();
    six.click();
    equals.click();

    MobileElement result = driver.findElement(By.className("android.widget.TextView"));
    System.out.println(result.getText());
}
```

Use External APK

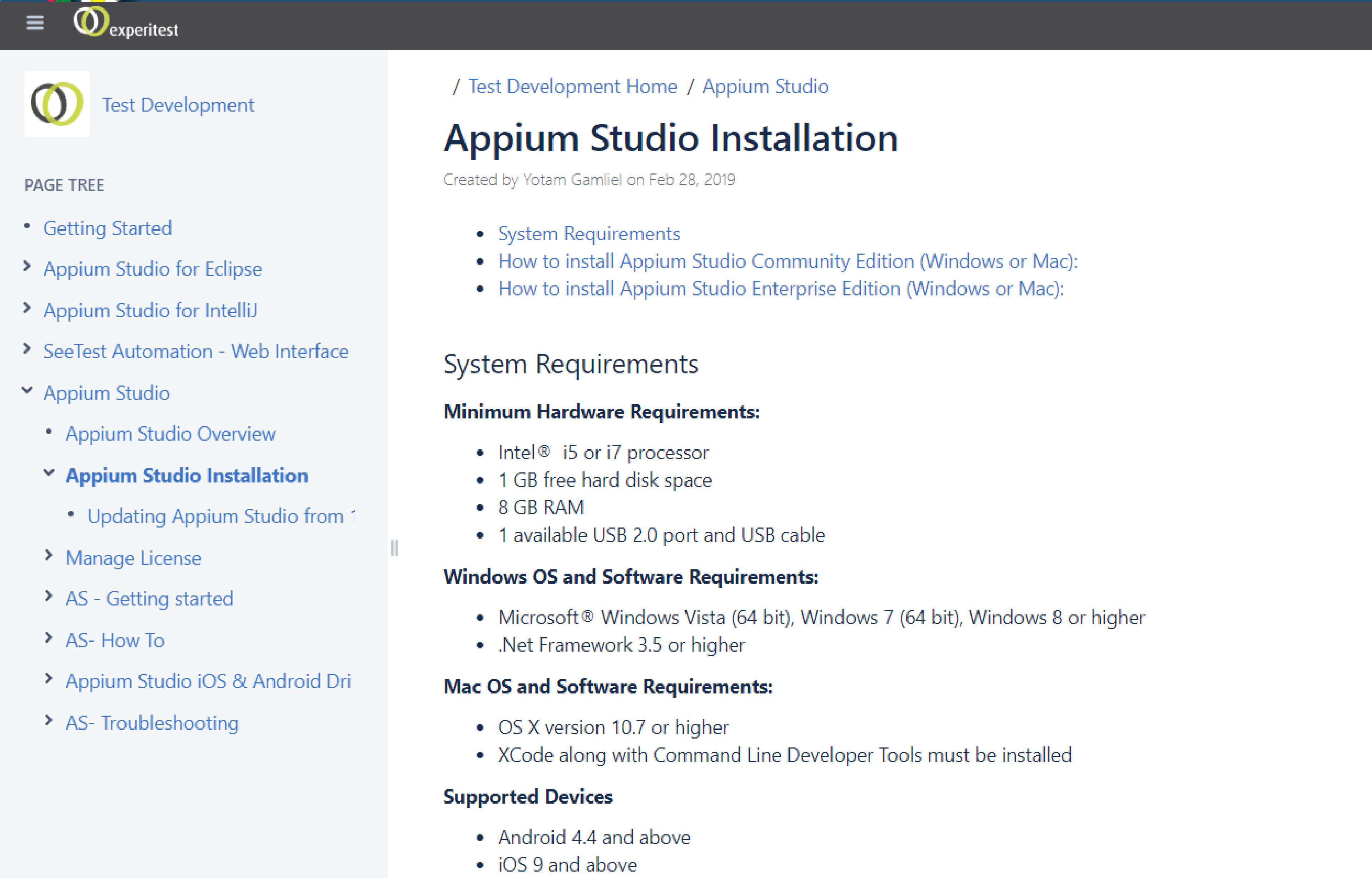


```
File app = new File("D:\\downloads\\Calculator_v7.8.apk");
DesiredCapabilities caps = new DesiredCapabilities();
caps.setCapability(MobileCapabilityType.PLATFORM_NAME, "Android");
caps.setCapability(MobileCapabilityType.PLATFORM_VERSION, "10");
caps.setCapability(MobileCapabilityType.UDID, "emulator-5554");
caps.setCapability(MobileCapabilityType.APP, app.getAbsolutePath());
```

Try it!

How to integrate with Jenkins and TestNG?!

Appium Studio



experitest
Test Development
PAGE TREE
Getting Started
Appium Studio for Eclipse
Appium Studio for IntelliJ
SeeTest Automation - Web Interface
Appium Studio
Appium Studio Overview
Appium Studio Installation
Updating Appium Studio from 1
Manage License
AS - Getting started
AS- How To
Appium Studio iOS & Android Dri
AS- Troubleshooting
/ Test Development Home / Appium Studio

Appium Studio Installation

Created by Yotam Gamliel on Feb 28, 2019

- System Requirements
- How to install Appium Studio Community Edition (Windows or Mac):
- How to install Appium Studio Enterprise Edition (Windows or Mac):

System Requirements

Minimum Hardware Requirements:

- Intel® i5 or i7 processor
- 1 GB free hard disk space
- 8 GB RAM
- 1 available USB 2.0 port and USB cable

Windows OS and Software Requirements:

- Microsoft® Windows Vista (64 bit), Windows 7 (64 bit), Windows 8 or higher
- .Net Framework 3.5 or higher

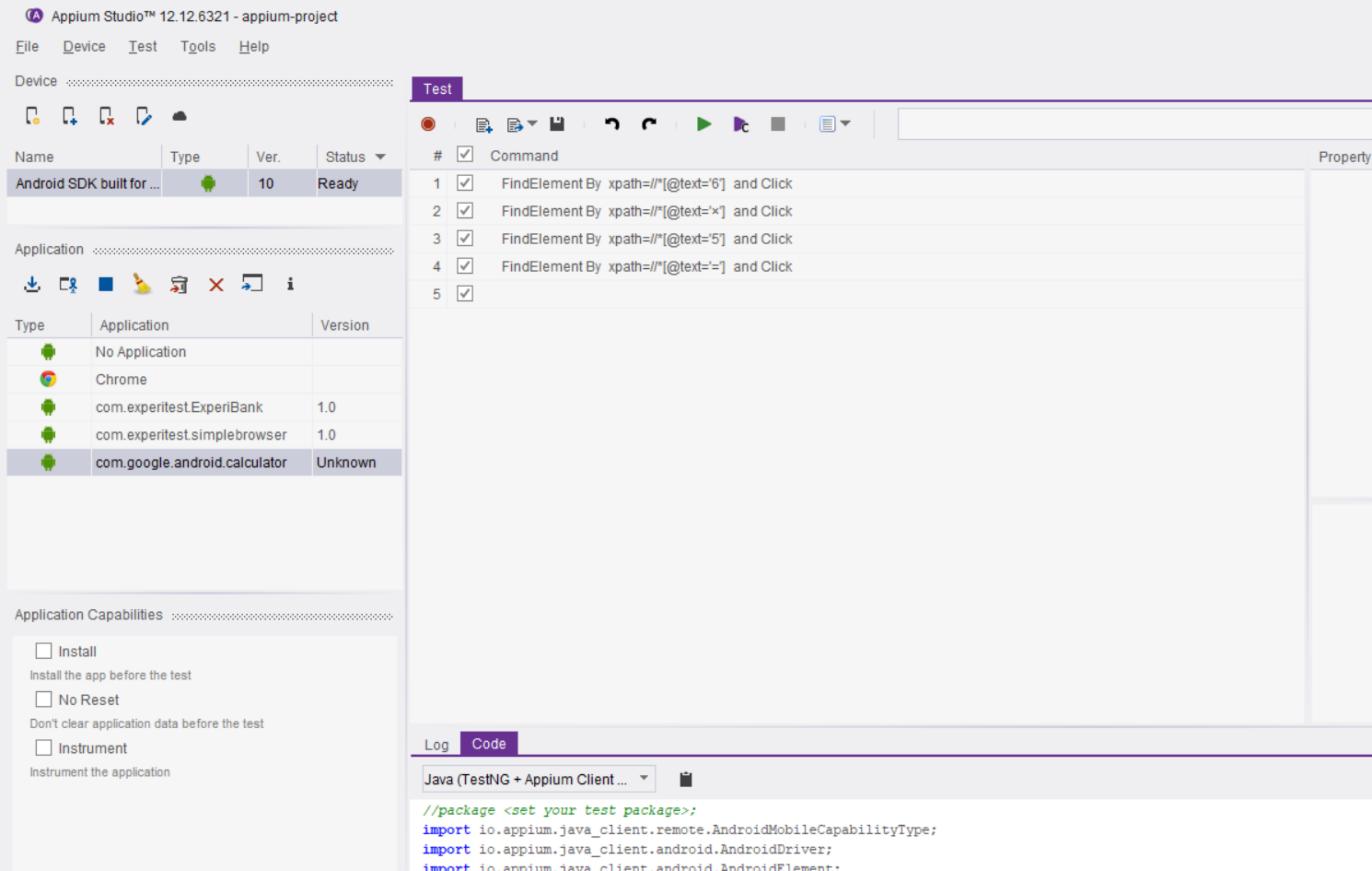
Mac OS and Software Requirements:

- OS X version 10.7 or higher
- XCode along with Command Line Developer Tools must be installed

Supported Devices

- Android 4.4 and above
- iOS 9 and above

Appium Studio



Name	Type	Ver.	Status
Android SDK built for ...		10	Ready

Type	Application	Version
	No Application	
	Chrome	
	com.experitest.ExperiBank	1.0
	com.experitest.simplebrowser	1.0
	com.google.android.calculator	Unknown

Application Capabilities

☐ Install
Install the app before the test

☐ No Reset
Don't clear application data before the test

☐ Instrument
Instrument the application

#	☑	Command
1	☑	FindElement By xpath=//*[@text='6'] and Click
2	☑	FindElement By xpath=//*[@text='×'] and Click
3	☑	FindElement By xpath=//*[@text='5'] and Click
4	☑	FindElement By xpath=//*[@text='='] and Click
5	☑	

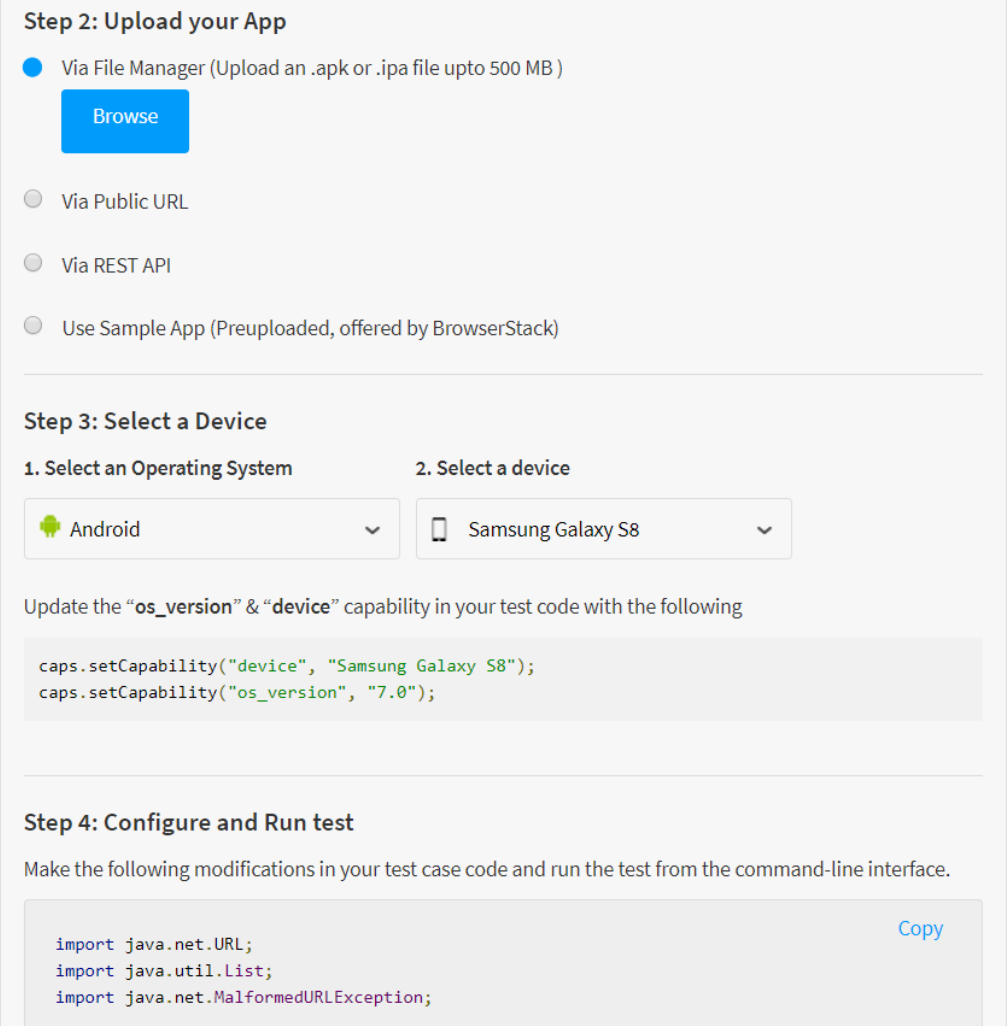
Java (TestNG + Appium Client ...

```
//package <set your test package>;
import io.appium.java_client.remote.AndroidMobileCapabilityType;
import io.appium.java_client.android.AndroidDriver;
import io.appium.java_client.android.AndroidElement;
```

Cloud Test

- SauceLabs
- BrowserStack
- Kobiton
- Experitest

BrowserStack Configuration



Step 2: Upload your App

● Via File Manager (Upload an .apk or .ipa file upto 500 MB)

○ Via Public URL

○ Via REST API

○ Use Sample App (Preuploaded, offered by BrowserStack)

Step 3: Select a Device

1. Select an Operating System: Android

2. Select a device: Samsung Galaxy S8

Update the "**os_version**" & "**device**" capability in your test code with the following

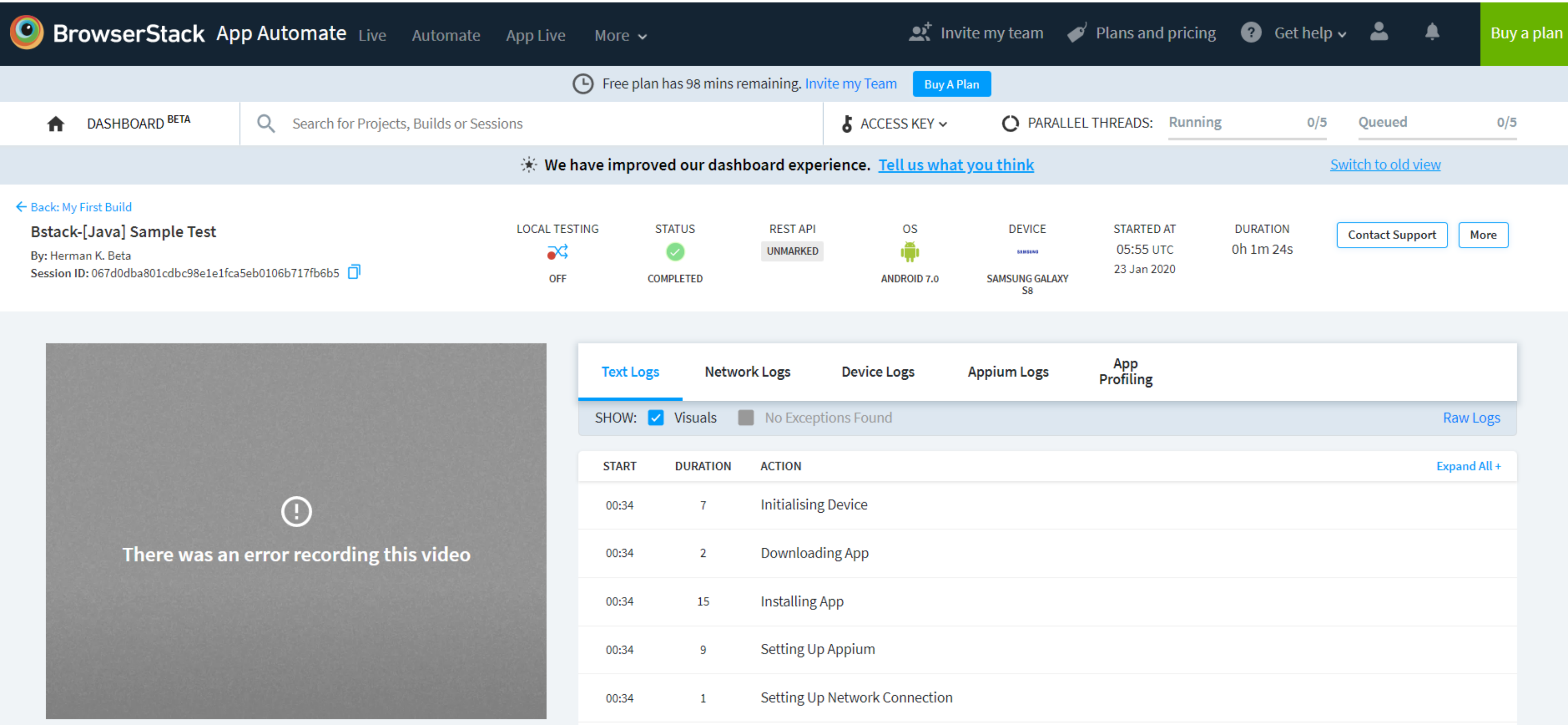
```
caps.setCapability("device", "Samsung Galaxy S8");
caps.setCapability("os_version", "7.0");
```

Step 4: Configure and Run test

Make the following modifications in your test case code and run the test from the command-line interface.

```
import java.net.URL;
import java.util.List;
import java.net.MalformedURLException;
```

BrowserStack Report



Free plan has 98 mins remaining. Invite my Team Buy A Plan

PARALLEL THREADS: Running 0/5 Queued 0/5

We have improved our dashboard experience. Tell us what you think

Switch to old view

← Back: My First Build

Bstack-[Java] Sample Test

By: Herman K. Beta

Session ID: 067d0dba801cdbc98e1e1fca5eb0106b717fb6b5

LOCAL TESTING	STATUS	REST API	OS	DEVICE	STARTED AT	DURATION
OFF	COMPLETED	UNMARKED	ANDROID 7.0	SAMSUNG GALAXY S8	05:55 UTC 23 Jan 2020	0h 1m 24s

There was an error recording this video

SHOW: ☑ Visuals ■ No Exceptions Found

START	DURATION	ACTION
00:34	7	Initialising Device
00:34	2	Downloading App
00:34	15	Installing App
00:34	9	Setting Up Appium
00:34	1	Setting Up Network Connection

Try it!

- Register to BrowserStack
- Create a new Project with sample apk