# Android Automation Using Appium

Sudheer

## Before Appium

If you use Apple's UIAutomation library without Appium you can only write tests using JavaScript .

Similarly, with Google's UiAutomator you can only write tests in Java. Appium opens up the possibility of true cross-platform native mobile automation. Finally!

# What is Appium

Appium is a mobile automation tool used for automating Android and iOS Mobile Applications.

Applications are of three types native,mobile web,hybrid.

Using Appium you can automate all the three types of applications.

For all latest updates on appium (like supported platforms, languages, and requirements).
https://github.com/appium/appium

# Features of Appium

Features of Appium:-

1. Open Source, and Its free.
2. Automates Android and IOS Applications.
3. Automates Native, Mobile Web , Hybrid Applications.
4. Cross Platform
   1. Test Android on OS X, Windows, Linux
   2. Test iOS on OS X
5. Supports any programming language.
6. Supports any framework.
7. You shouldn't have to recompile your app or modify it in any way in order to automate it.

# Limitations of Appium

**Limitations**

- Android
  - No Support for Toast Messages
  - Android Version 4.2+ required
- iOS
  - Needs mac OSX 10.7+, lower versions not supported

# Native, Mobile Web,Hybrid

- Native apps are developed by using native libraries(API's) of a specific platform, hence these are developed specifically for one platform, and can take full advantage of all the device features (camera, gps,etc.)
  - These applications are downloaded from plat specific app store.
- Mobile web apps are web apps accessed using a mobile browser
  - Are developed by using standard web technologies HTML5, JavaScript, CSS.
  - Appium supports Safari browser on iOS
  - Chrome or the built-in ‘Browser’ on Android.
- Hybrid apps is a combination of web and native.

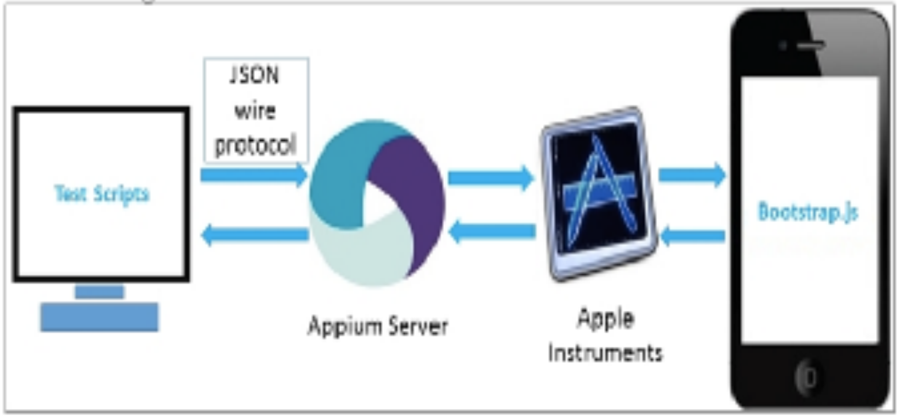
# Appium Architecture

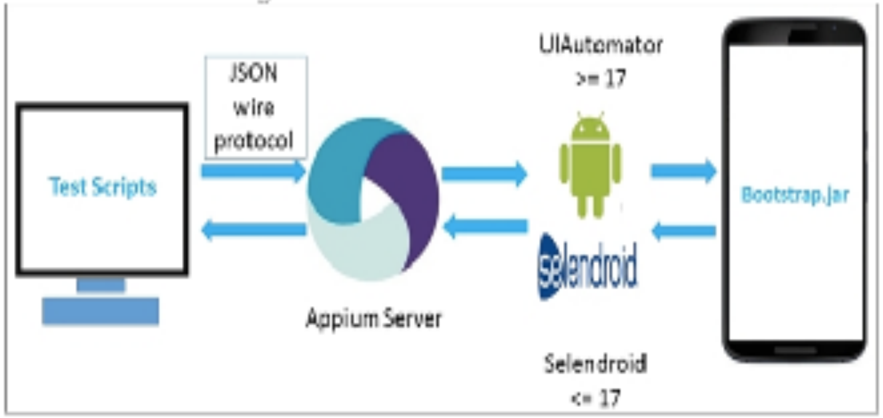
IOS:-
On an iOS device, Appium uses Apple's UIAutomation API to interact with the UI elements. UIAutomation is a JavaScript library provided by Apple to write test scripts; Appium utilizes these same libraries to automate iOS apps

When we execute the test scripts, it goes in the form of JSON through an HTTP request to the Appium server. The Appium server sends the command to the instruments, and the instruments look for the bootstrap.js file, which is pushed by the Appium server to the iOS device. Then, these commands execute in the bootstrap.js file within the iOS instruments' environment. After the execution of the command, the client sends back the message to the Appium server with the log details of the executed command.

**Android:-**
On an Android device, Appium uses the UIAutomator framework to automate the apps. UIAutomator is a framework that is developed by the Android developers to test the Android user interface.

we have a UIAutomator/Selendroid in place of Apple instruments and bootstrap.jar in place of the bootstrap.js file.
Appium supports Android versions greater than or equal to 17; for earlier versions, it uses the Selendroid framework. When we execute the test scripts, Appium sends the command to the UIAutomator or Selendroid on the basis of the Android version.
Here, bootstrap.jar plays the role of a TCP server, which we can use to send the test command in order to perform the action on the Android device using UIAutomator/Selendroid.





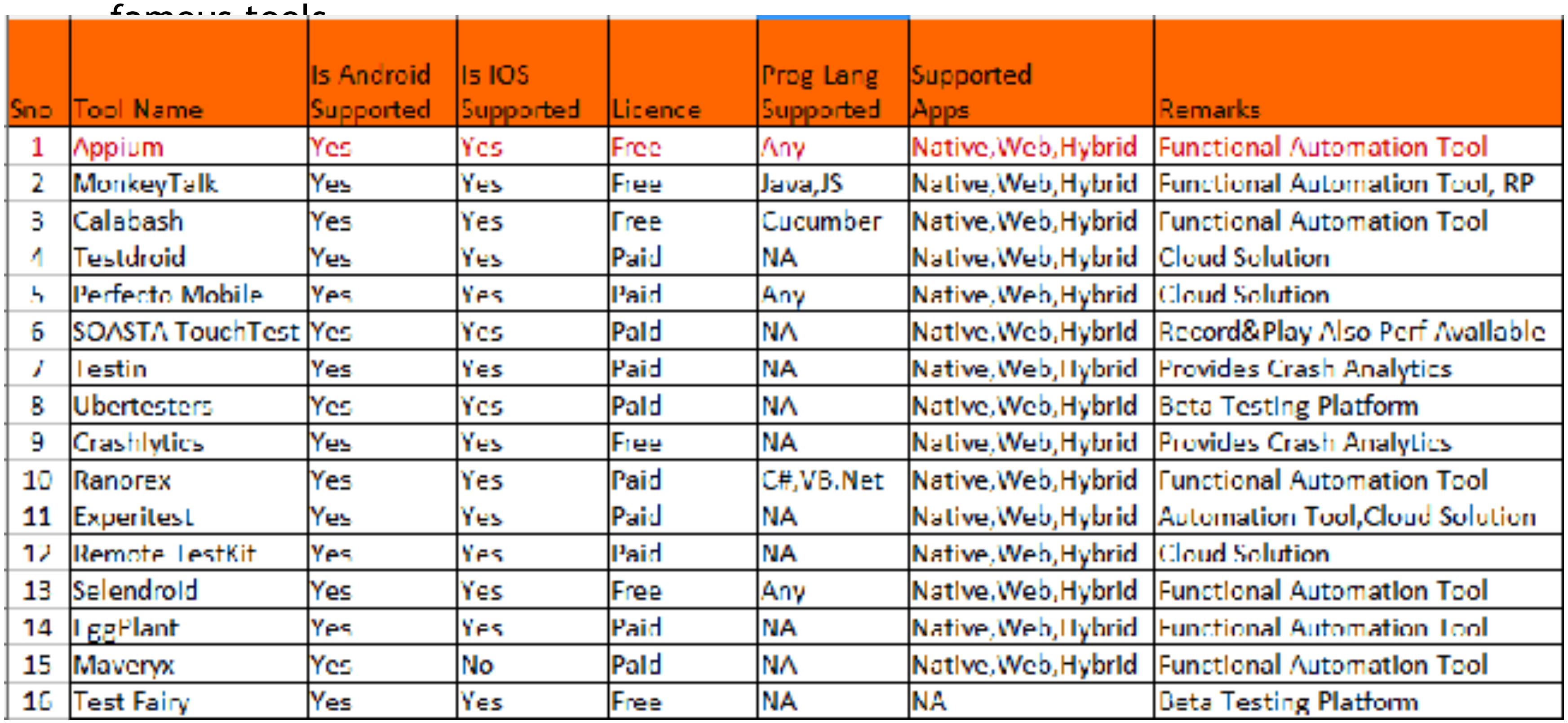
## Challenges

Identifying mobile automation tools is more challenging comparatively desktop or web application due to following aspects:

- Variant sizes of mobile phones.
- Variant Mobile platform versions (Android 4.2,4.4, and IOS 7,8,etc.)
- Variant device configurations.
- Mobile applications have smaller foot print on virtual machine compared to desktop/web applications.

# Mobile tools availability

- Desktop/web applications tools are not used for  mobile application testing.
- Hence a complex scripting technique and new tool development required for mobile application testing, some of the standard automation tools for mobile app testing.
- Among all the listed tools, **Appium, Monkey Talk** , Selendroid and IOSDriver are famous tools



| Sno | Tool Name | Is Android Supported | Is IOS Supported | Licence | Prog Lang Supported | Supported Apps | Remarks |
|---|---|---|---|---|---|---|---|
| 1 | Appium | Yes | Yes | Free | Any | Native,Web,Hybrid | Functional Automation Tool |
| 2 | MonkeyTalk | Yes | Yes | Free | Java,JS | Native,Web,Hybrid | Functional Automation Tool, RP |
| 3 | Calabash | Yes | Yes | Free | Cucumber | Native,Web,Hybrid | Functional Automation Tool |
| 4 | Testdroid | Yes | Yes | Paid | NA | Native,Web,Hybrid | Cloud Solution |
| 5 | Perfecto Mobile | Yes | Yes | Paid | Any | Native,Web,Hybrid | Cloud Solution |
| 6 | SOASTA TouchTest | Yes | Yes | Paid | NA | Native,Web,Hybrid | Record&Play Also Perf Available |
| 7 | Testin | Yes | Yes | Paid | NA | Native,Web,Hybrid | Provides Crash Analytics |
| 8 | Ubertesters | Yes | Yes | Paid | NA | Native,Web,Hybrid | Beta Testing Platform |
| 9 | Crashlytics | Yes | Yes | Free | NA | Native,Web,Hybrid | Provides Crash Analytics |
| 10 | Ranorex | Yes | Yes | Paid | C#,VB.Net | Native,Web,Hybrid | Functional Automation Tool |
| 11 | ExperiTest | Yes | Yes | Paid | NA | Native,Web,Hybrid | Automation Tool,Cloud Solution |
| 12 | Remote TestKit | Yes | Yes | Paid | NA | Native,Web,Hybrid | Cloud Solution |
| 13 | Selendroid | Yes | Yes | Free | Any | Native,Web,Hybrid | Functional Automation Tool |
| 14 | EggPlant | Yes | Yes | Paid | NA | Native,Web,Hybrid | Functional Automation Tool |
| 15 | Maveryx | Yes | No | Paid | NA | Native,Web,Hybrid | Functional Automation Tool |
| 16 | Test Fairy | Yes | Yes | Free | NA | NA | Beta Testing Platform |

# Requirement for Android Application Automation using Appium

- PlatForm:-
  - Windows 7 or later
  - Mac OS X 10.7 or later
  - Linux

- Software's:-
  - Java
  - Android SDK (Version 17 or later)
  - Appium
  - Eclipse

- JAR files to be downloaded:-
  - Selenium WebDriver Java client
  - Appium Java Client
  - GSON (is part of selenium jars)

# Requirement for iOS Application Automation using Appium

- PlatForm:-
  - Mac OS X 10.7 or later
- Software's:-
  - Java
  - Appium
  - Eclipse
  - **XCODE**
  - Homebrew - used for install any s/w that apple doesn't provide.
  - Node and npm - used for installing appium from command line
- JAR files to be downloaded:-
  - Selenium WebDriver Java client
  - Appium Java Client
  - GSON (is part of selenium jars)

# Download required jar files

- **Selenium Server and Web-Driver Java client**
  - **Download latest selenium version from**
  - **http://www.seleniumhq.org/download/**
  - do not download beta version
- **Appium Java client**
  - http://search.maven.org/#search|ga|1|appium%20java%20client
  - Download latest version at that momemnt "java-client-4.1.2.jar"
- **Gson**
  - http://mvnrepository.com/artifact/com.google.code.gson
  - Its part of selenium jars no need to download seperately.

# Installing Android SDK

- Windows
  - go to this page [https://developer.android.com/studio/index.html](https://developer.android.com/studio/index.html)
  - Scroll down, and click on “Download options”
  - Scroll down until you see “Get just the command line tools”
  - Now download the software as per the OS.
  - for ex (if windows - “tools_r25.2.3-windows.zip”
  - Once extracted you need to set environment variables
    - ANDROID_HOME
    - set path variable

# Settingup SDK Manager

- Once Adnroid SDK is  downloaded and configured, Need to setup SDK Manager.
- Go to Android SDK Location, In my system the location is
  - C:\Users\deepak\AppData\Local\Android\sdk
- Open **SDK Manager.exe**
  - which inside **tools/bin** folder
- Tick on **Tools** to install it.
- And based on emulator/real device  version that you want to use for automation running, install required components.
  - For ex, If you want launch emulator with plat form version 6 , then choose that component for install.

## Installing Appium

- Windows
  - Download from http://appium.io/
  - Extract the zip that is downloaded.
  - Launch the .exe file.
  - Follow the setup wizard to install appium server.
  - That's it!

## Installing Eclipse

- Windows
  - Download from https://eclipse.org/
  - Launch the .exe file you downloaded.
  - Follow the setup wizard to install eclipse.
  - That's it!

# iOS - Install Homebrew & Node,NPM

all these for command line execution

Home-brew:-

- run the below command in the terminal
    - ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)”
- follow the instructions on terminal
- execute the command after installation done on terminal
    - brew doctor
- fix any issues if results of command “brew doctor” shows.

Node&NPM

- run the below command in the terminal
    - brew install node

## iOS - Install Homebrew & Node,NPM , Continuation ….

### Quick Start

Kick up an Appium server, and then run a test written in your favorite WebDriver-compatible language! You can run an Appium server using node.js or using the application, see below.

#### Using Node.js

```
$ npm install -g appium
$ appium
```

As we said above, you may want to run `appium-doctor` to ensure your system is set up properly:

```
$ npm install -g appium-doctor
$ appium-doctor
$ appium-doctor - - iOS
$ appium-doctor - - android
```

## iOS - Intalling XCODE

- Open AppStore in mac
    - *command* + *space* and enter App Store
- Search for *Xcode*
- You see list of applications, click on *Xcode*
- click on install
- Xcode previous version downloads,
    - https://developer.apple.com/download/more/

## Show installed SDKs for iOS

```
dhcp-india-vpn-vpnpool-10-191-197-146:~ dnreddy$ xcodebuild -showsdks
iOS SDKs:
	iOS 10.2                      	-sdk iphoneos10.2

iOS Simulator SDKs:
	Simulator - iOS 10.2          	-sdk iphonesimulator10.2

macOS SDKs:
	macOS 10.12                   	-sdk macosx10.12

tvOS SDKs:
	tvOS 10.1                     	-sdk appletvos10.1

tvOS Simulator SDKs:
	Simulator - tvOS 10.1         	-sdk appletvsimulator10.1

watchOS SDKs:
	watchOS 3.1                   	-sdk watchos3.1

watchOS Simulator SDKs:
	Simulator - watchOS 3.1       	-sdk watchsimulator3.1
```

Note:-if the command not given any results, it means that you need to install Xcode command line tools.

## Installing Android Application - APK Info

- Real device - install APK info from play store.
  - https://play.google.com/store/apps/details?id=de.migali.soft.apkinfo&hl=en
- Emulator - download apk file using following link
  - https://apkpure.com/
  - and install using following command from terminal,
  - *adb install "/Users/dnreddy/Documents/Appium/Apk Info free_v1.0.2.4.F_apkpure.com.apk"*

# Finding Locators information of WebElements

iOS

| | | |
|---|---|---|
| - Native | - | Appium Inspector |
| **- Web** | **-** | **Safari Browser** |

**Android**

| | | |
|---|---|---|
| **- Native** | **-** | **UIAutomatorViewer, [Appium Inspector]** |
| - Web | - | Chrome ADB Plugin |

## Identifying elements on Android

- UIAutomatorviewer: using which you can able to identify the elements of native application.
- Chrome ADB Plugin: using which you can able to identify the elements of web application on mobile browser.

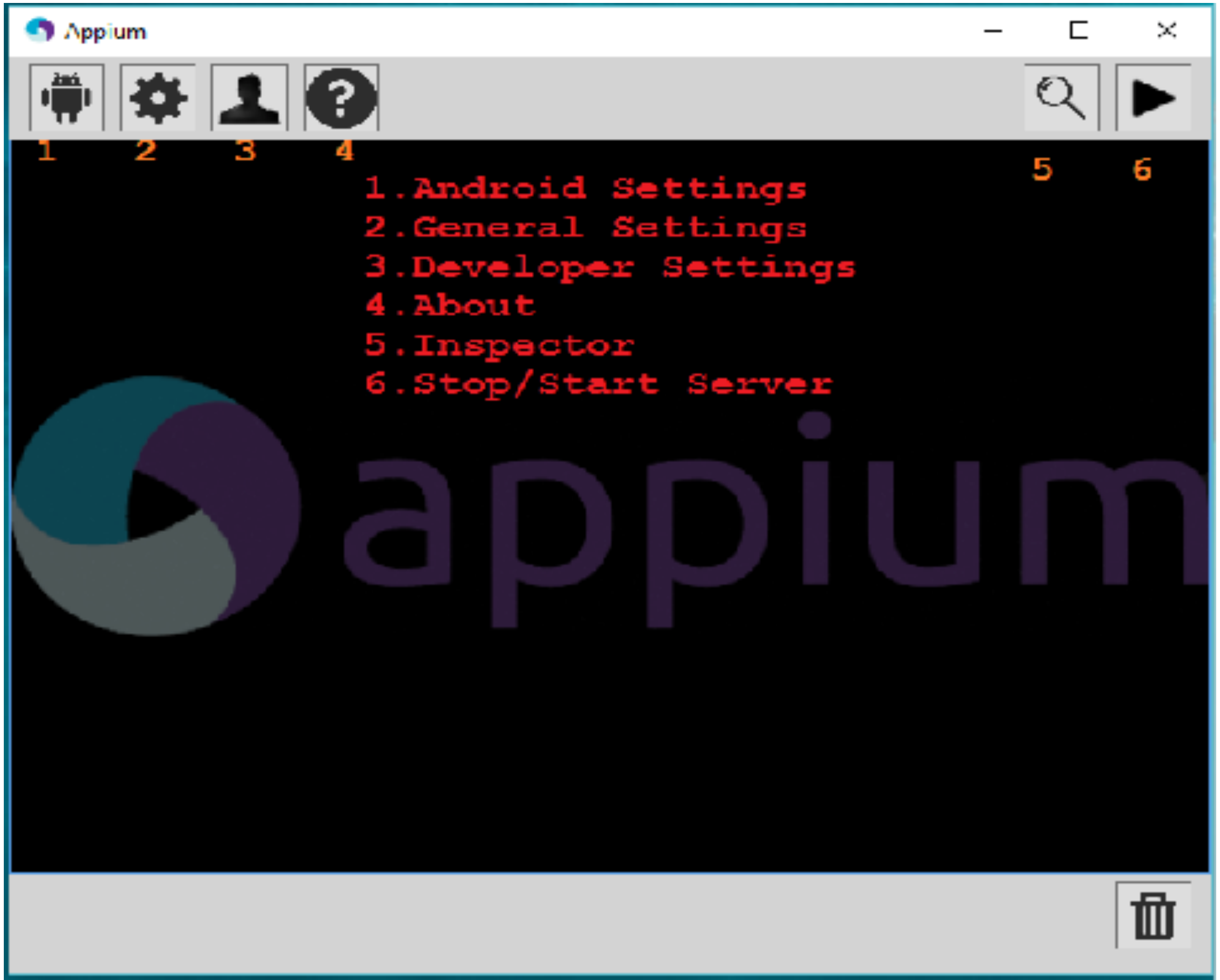
## Finding web elements in iOS - Web Application

- Enable WebInspector in the IOS Device/Simulator
  - Settings application->Safari->Advance->WebInspector
- Open Safari browser on iOS Device/Simulator and open the application url where you to find the elements
  - for example:- www.google.com
- Open safari browser on mac
- In the Menu->Develop-> Choose Simulaor/Real Device -> Choose application running on iOS Device.
- It opens a Web Inspector, “start element selection” (this icon looks like location icon).
- Then move cursor to the element that you want to identify in the Application opened in iOS device.
- Then in the safari window the corresponding html tag will be displayed.

# Finding elements for Native iOS Application

- Launch appium with the following options
  - Close iOS Simulator if you have already opened
  - If its real device unplug and plug
  - iOS Settings
    - App Path
    - Check Force Device and Device Name
    - Platform Version
  - General Settings
    - Check PreLaunch Application
    - Overriding existing sessions

# Appium Widown UI



1. Android Settings
2. General Settings
3. Developer Settings
4. About
5. Inspector
6. Stop/Start Server

## Automation Fundamentals

Any automation tools works on the following fundamentals, whether the automation is appium or selenium or qtp or etc.

1.How to identify/locate an element on the screen.
2.How do you perform an action on the identified element.

*elements on the screen* - button,text field, checkbox,dropdown, radio button,etc.

In Appium these elements are called *WebElements*.

ex:- click on 8 , +, 7 , =

# Appium Locators

Are used to locate/identify an element

1. Name
2. Id
3. ClassName
4. TagName
5. AccessibilityID
6. XPath
7. CSS
8. LinkText
9. PartialLinkText

**For Native Apps:-**
Text - Name
resource-id - Id
content-desc - AccessibilityId

## Create a Project and Write test

- Create a java project on eclipse
- Add the libraries to build path of java project
  - Selenium Java libraries
  - Selenium Standalone server
  - Appium Java Client
- Set Desired Capabilities
- Create Android Driver (or IOS Driver).
- Find Elements, and Perform Actions
- Close App

## Necessary desired capabilities for Android

1. Click on the **Android Settings** icon and set / from code directly.
2. Select **Application path** and provide the path of the application.
3. Select **Package** and choose it from the drop-down.
4. Select **Launch Activity** and choose an activity from the drop-down.
5. Select **Launch AVD** and choose a created emulator from the list.
6. Select **PlatformVersion** from the drop-down menu.
7. Select **Device Name** and type *Android emulator*.
8. Select **Automation Name** appium
9. Now, start the Appium Server.

Note:-

- If App is already installed 2-4 option not required, and set those package, and launch activity options directly on Android Desired Capabilities.
- Also MobileCapabilityType.APP no need to set Desired capabilities.

# Test AndroidNativeApp - Real Device

```
DesiredCapabilities caps = new DesiredCapabilities();
caps.setCapability(MobileCapabilityType.PLATFORM_NAME, "Android");
caps.setCapability(MobileCapabilityType.PLATFORM_VERSION, "4.4.2");
caps.setCapability(MobileCapabilityType.DEVICE_NAME, "B1-730HD");
caps.setCapability(AndroidMobileCapabilityType.APP_PACKAGE, "com.android.calculator2");
caps.setCapability(AndroidMobileCapabilityType.APP_ACTIVITY, "com.android.calculator2.Calculator");

AndroidDriver driver = new AndroidDriver(new URL("http://127.0.0.1:4723/wd/hub"),caps);
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

WebElement element8 = driver.findElement(By.name("8"));
element8.click();

WebElement elementPlus = driver.findElement(By.id("com.android.calculator2:id/plus"));
elementPlus.click();

WebElement element7 = driver.findElement(By.id("digit7"));
element7.click();

WebElement equalsElement = driver.findElementByAccessibilityId("equals");
equalsElement.click();
```

# Test Android Web App

```
DesiredCapabilities caps = new DesiredCapabilities();
caps.setCapability(MobileCapabilityType.PLATFORM_NAME, "Android");
caps.setCapability(MobileCapabilityType.PLATFORM_VERSION, "5.0.2");
caps.setCapability(MobileCapabilityType.DEVICE_NAME, "Android emulator");
caps.setCapability("avd", "Nexus_4_API_21");
caps.setCapability(MobileCapabilityType.BROWSER_NAME, "Browser"); //"chrome"


//Create Android Driver
AndroidDriver driver = new AndroidDriver(new URL("http://127.0.0.1:4723/wd/hub"), caps);
driver.manage().timeouts().implicitlyWait(100, TimeUnit.SECONDS);

driver.get("https://www.google.com");
//Enter "appium" on the text field
WebElement textElement = driver.findElement(By.id("lst-ib"));
textElement.clear();
textElement.sendKeys("Appium");

driver.close();
```

# Test Android Hybrid App

```
DesiredCapabilities caps = new DesiredCapabilities();
caps.setCapability(MobileCapabilityType.PLATFORM_NAME, "Android");
caps.setCapability(MobileCapabilityType.PLATFORM_VERSION, "5.0.2");
caps.setCapability(MobileCapabilityType.DEVICE_NAME, "Android emulator");
caps.setCapability("avd", "Nexus_4_API_21");

caps.setCapability(AndroidMobileCapabilityType.APP_PACKAGE, "com.example.testapp");
caps.setCapability(AndroidMobileCapabilityType.APP_ACTIVITY, "com.example.testapp.MainActivity");

AndroidDriver driver = new AndroidDriver(new URL("http://127.0.0.1:4723/wd/hub"), caps);
driver.manage().timeouts().implicitlyWait(100, TimeUnit.SECONDS);
//Enter text(url) on the text field
WebElement urlElement = driver.findElement(By.id("urlField"));
urlElement.sendKeys("https://www.google.com");

//Click on Go Button
WebElement goButton = driver.findElement(By.name("Go"));
goButton.click();

//Switching to WebApp from Native
Set<String> contextHandles = driver.getContextHandles();
for (String id : contextHandles)
{
   System.out.println(id);  //NATIVE_APP, WEBVIEW_com.example.testapp
}

driver.context("WEBVIEW_com.example.testapp");

//Its going to launch google.com application, want to enter a text called "Appium"
WebElement textElement = driver.findElement(By.id("lst-ib"));
textElement.clear();
textElement.sendKeys("Appium");

driver.closeApp();
```

# Test AndroidNativeApp - Without PackageName and Activity

```
DesiredCapabilities caps = new DesiredCapabilities();
File file = new File("/Users/dnreddy/Documents/Appium/BMICalculator.apk");
caps.setCapability(MobileCapabilityType.APP, file);
caps.setCapability(MobileCapabilityType.PLATFORM_NAME, "Android");
caps.setCapability(MobileCapabilityType.PLATFORM_VERSION, "4.4.2");
caps.setCapability(MobileCapabilityType.DEVICE_NAME, "B1-730HD");

AndroidDriver driver = new AndroidDriver(new URL("http://127.0.0.1:4723/wd/hub"),caps);
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

List<WebElement> textElement = driver.findElements(By.className("android.widget.EditText"));
textElement.get(0).sendKeys("Sudheer");


List<WebElement> buttonElement = driver.findElements(By.className("android.widget.Button"));
buttonElement.get(0).click();

WebElement metricElement = driver.findElement(By.name("Metric"));
metricElement.click();
```

# Test AndroidNativeApp - Emulator

```
DesiredCapabilities caps = new DesiredCapabilities();
caps.setCapability(MobileCapabilityType.PLATFORM_NAME, "Android");
caps.setCapability(MobileCapabilityType.PLATFORM_VERSION, "5.0.2");
//if its real device, give actual device name
caps.setCapability(MobileCapabilityType.DEVICE_NAME, "Android emulator");
caps.setCapability("avd", "Nexus_4_API_21"); //is not required if its real device
//Identifiying application using desired capabilities - for android
caps.setCapability(AndroidMobileCapabilityType.APP_PACKAGE, "com.android.calculator2");
caps.setCapability(AndroidMobileCapabilityType.APP_ACTIVITY, "com.android.calculator2.Calculator");

//Create Android Driver
AndroidDriver driver = new AndroidDriver(new URL("http://127.0.0.1:4723/wd/hub"), caps);
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

WebElement digit8 = driver.findElement(By.id("com.android.calculator2:id/digit_8"));
digit8.click();

WebElement plusElement = driver.findElementByAccessibilityId("plus");
plusElement.click();

WebElement digit7 = driver.findElement(By.name("7"));
digit7.click();

WebElement eqElement = driver.findElement(By.id("eq"));
eqElement.click();

driver.closeApp();
```

# Important Contacts - Android

**open terminal/command prompt and type the following commands,**

```
android          - SDK Manager
android avd      - AVD Manager
```

how to start android emulator from command prompt?

```
emulator -avd Nexus_4_API_21 -netspeed full -netdelay none -http-proxy http://www-proxy.us.oracle.com:80
```

How to Use ADB to Control Keypress events on Android?

```
adb shell /system/bin/input keyevent 1   //KEYCODE_MENU
```

Fore more constants:- http://developer.android.com/reference/android/view/KeyEvent.html

List of important adb commands?

```
adb devices //to see the list of devices
adb -e install <app_path>    //to install a apk
adb uninstall <packagename of app>
adb -s emulator-5556 install helloWorld.apk  //to install apk on a specified connected device
adb shell dumpsys activity activities //to get the list of activities on the opened application
adb pull <remote> <local>
adb push <local> <remote>  //adb push foo.txt /sdcard/foo.txt
adb kill-server //terminate the the ‘adb’ server process
adb start-server //start the ‘adb’ server process if its not started already
adb wait-for-device //block the execution until the device is online
```

fore more information :- http://developer.android.com/tools/help/adb.html

# Create iOS Simulator

Commad+space - it opens search bar
Search for “Simulator”
Launch iOS Simulator

or
Open XCODE
Xcode -> Open Developer Tools -> Simulator

Choose Another Device
Menu—Hardware—Devices—choose platform versions -> iphone simulator model

Create new simulator
Menu—Hardware—Devices—Manage Devices
At the bottom — click on + button — select Simulator
Enter Device Name, Choose Device Type, Select OS Version

Create new platform versions incase if the versions is not available
Menu—Hardware—Devices—Manage Devices
At the bottom — click on + button — select Simulator
Choose Device Type, Select OS Version as download more simulators
select the iOS versions, and click on it for download.

Note:-
-If desired version is not available in the drop down , click on “Download Simulator”
-click on the download button of Desire Version, and installed
- after download and install of the version, it will be displayed in the version drop down.
- choose the latest version that is downloaded.

Click on create button , to create simulator.

,

# iOS Simulator

## How to open a simulator in iOS?

1.click on the below keys in the keyboard
command+space = which opens a search
2.Type “simulator” select simulator in the search result , and click on it.

## 2.How to see list of simulator?

1.open the simulator
2.On menu displayed on the top of window choose
Hardware -> Device -> Manage Devices..
3.Then you can see list of Simulators available

## Create AVD

- Now open **AVD Manager.exe**
- Click on **create button on** right hand side to launch AVD Create screen.
- Fill the required details, and create AVD.
- Alternative way to open the AVD Manager is,
  - type “android avd" in command prompt

# Android Emulator

How to open a simulator in Emulator In Android?

How to open Emulator in Android?

1. Open command prompt type the following
   android ad
2. It opens a windows , thats shows list of emulators available
3. Choose one of the emulator, and click “start”
4. If emulators are shown, you can create a new emulator.

# ADB Plugin

## How to install ADB Plugin:

open the chrome browser
click on settings icon on the top right corner( it will be in 3 vertical dots).
Click on Extension, scroll down the page
Click on get more extension link
search for “ADB plugin”
Click on ADB plugin in the search result, and install it.

## Inspect WebElement on Chrome Browser:

1.Open the browser
2.Load the Application that you want to inspect
3.Open the chrome browser
4.Click on ADB plugin (make sure you have already installed it).
5.It shows “chrome” under that the application launched, now click on “inspect” in the bottom.
6.It opens a another page with Application screen shot launched.
7.Select the icon “inspect” and inspect the web elements.

## Appium Inspector for Identifying the locators of Android Native Application

- Open the Appium Server
- Click on the Android icon, to open the Android settings
- Choose the App Path
- Choose Package,Wait for Package,Activity Name
- Close Android settings after completing
- Now Click on General settings icon, to open general settings
- Select the Prelaunch Application
- Now Launch the server
- App will be installed on the connected device/emulator
- and app will be launched on the device.
- Now click on the Appium inspector, which loads the app screen with app elements locators.

# Advanced UI Actions

- Mobile gestures
  - long press
  - drag and drop
- Handling alerts
- spinners
- drop downs
- switch button
- slide seek bar
- Swipe
- Zoom
- Capturing screenshots
- Capturing screenshots on test failure
- scroll up or scroll down

TouchAction
MultiTouchAction classes

```
TouchAction tAction=new TouchAction(driver);
  tAction.longPress(WebElement);
  tAction.perform();
```

# Advanced UI Actions - Sample Code

Scroll to

```
driver.scrollTo(“Nitika");
```

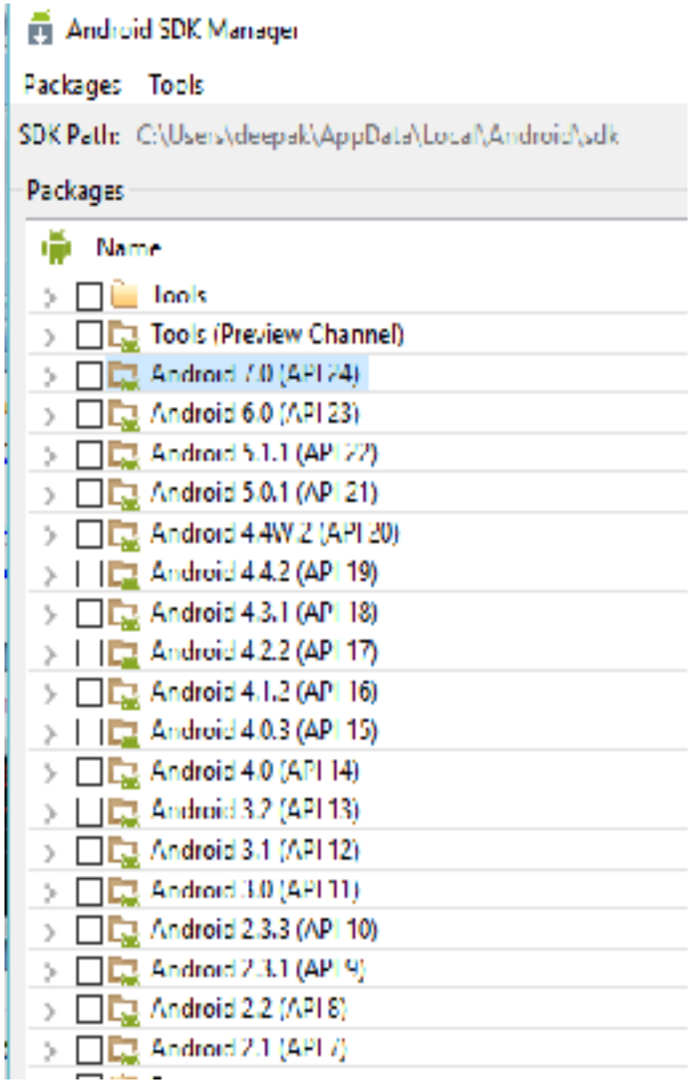
Long Press

```
TouchAction tAction=new TouchAction(driver);
tAction.longPress(webElementObj).perform();
```

**Drag N Drop**

```
WebElement appsIcon=driver.findElementByAccessibilityId("Apps");
appsIcon.click();
WebElement calculator=driver.findElementByName(“Calculator");
TouchAction action=new TouchAction(driver);
action.press(calculator);
action.moveTo(driver.findElement(By.name("App info"))).release().perform();
```

# Android Platform version vs API Level



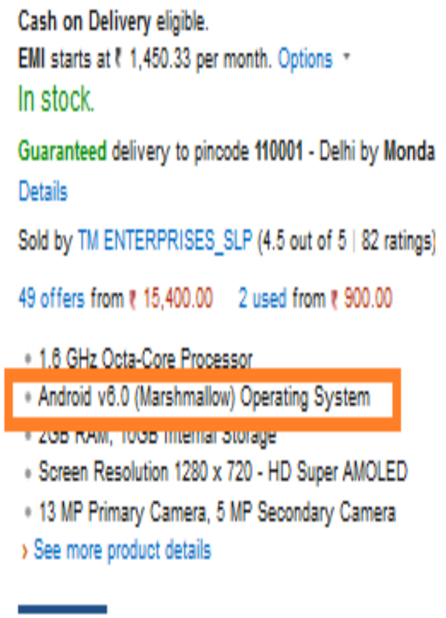
Android SDK Manager

Packages Tools

SDK Path: C:\Users\deepak\AppData\Local\Android\sdk

Packages

| Name |
|---|
| Tools |
| Tools (Preview Channel) |
| Android 7.0 (API 24) |
| Android 6.0 (API 23) |
| Android 5.1.1 (API 22) |
| Android 5.0.1 (API 21) |
| Android 4.4W.2 (API 20) |
| Android 4.4.2 (API 19) |
| Android 4.3.1 (API 18) |
| Android 4.2.2 (API 17) |
| Android 4.1.2 (API 16) |
| Android 4.0.3 (API 15) |
| Android 4.0 (API 14) |
| Android 3.2 (API 13) |
| Android 3.1 (API 12) |
| Android 3.0 (API 11) |
| Android 2.3.3 (API 10) |
| Android 2.3.1 (API 9) |
| Android 2.2 (API 8) |
| Android 2.1 (API 7) |



Cash on Delivery eligible.
EMI starts at ₹ 1,450.33 per month. Options
In stock.
Guaranteed delivery to pincode 110001 - Delhi by Monda
Details
Sold by TM ENTERPRISES_SLP (4.5 out of 5 | 82 ratings)
49 offers from ₹ 15,400.00 2 used from ₹ 900.00

- 1.6 GHz Octa-Core Processor
- Android v6.0 (Marshmallow) Operating System
- 2GB RAM, 16GB Internal Storage
- Screen Resolution 1280 x 720 - HD Super AMOLED
- 13 MP Primary Camera, 5 MP Secondary Camera

› See more product details

Samsung Galaxy J7

## SampleTest for Native Android App

SampleTest.java

# Appium 3.4.0 with XCODE8.2

```
Exception in thread "main" org.openqa.selenium.WebDriverException: An unknown server-side error occurred while processing the command.
Original error: Platform version must be 9.3 or above. '8.4' is not supported. (WARNING: The server did not provide any stacktrace
information)
Command duration or timeout: 14.73 seconds
Build info: version: '2.53.1', revision: 'a36b8b1cd5757287168e54b817830adce9b0158d', time: '2016-06-30 19:26:09'
System info: host: 'dhcp-india-vpn-vpnpool-10-191-197-146.vpn.oracle.com', ip: '10.191.197.146', os.name: 'Mac OS X', os.arch: 'x86_64',
os.version: '10.12.2', java.version: '1.8.0_101'
Driver info: io.appium.java_client.ios.IOSDriver
	at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
	at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
	at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
	at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
	at org.openqa.selenium.remote.ErrorHandler.createThrowable(ErrorHandler.java:206)
	at org.openqa.selenium.remote.ErrorHandler.throwIfResponseFailed(ErrorHandler.java:158)
	at org.openqa.selenium.remote.RemoteWebDriver.execute(RemoteWebDriver.java:678)
	at io.appium.java_client.DefaultGenericMobileDriver.execute(DefaultGenericMobileDriver.java:42)
	at io.appium.java_client.AppiumDriver.execute(AppiumDriver.java:1)
	at io.appium.java_client.ios.IOSDriver.execute(IOSDriver.java:1)
	at org.openqa.selenium.remote.RemoteWebDriver.startSession(RemoteWebDriver.java:249)
	at org.openqa.selenium.remote.RemoteWebDriver.<init>(RemoteWebDriver.java:131)
	at org.openqa.selenium.remote.RemoteWebDriver.<init>(RemoteWebDriver.java:144)
	at io.appium.java_client.DefaultGenericMobileDriver.<init>(DefaultGenericMobileDriver.java:37)
	at io.appium.java_client.AppiumDriver.<init>(AppiumDriver.java:162)
	at io.appium.java_client.AppiumDriver.<init>(AppiumDriver.java:171)
	at io.appium.java_client.ios.IOSDriver.<init>(IOSDriver.java:56)
	at SampleTestForNativeApplicationInIOS.main(SampleTestForNativeApplicationInIOS.java:26)
```

```
caps.setCapability(MobileCapabilityType.APP, file);
caps.setCapability(MobileCapabilityType.PLATFORM_NAME, "iOS");
caps.setCapability(MobileCapabilityType.PLATFORM_VERSION, "8.4");
caps.setCapability(MobileCapabilityType.DEVICE_NAME, "iPhone 6");
caps.setCapability(MobileCapabilityType.AUTOMATION_NAME, "XCUITest");
```

# Appium Desktop 1.0.0 with XCODE8.2

```
Exception in thread "main" org.openqa.selenium.SessionNotCreatedException: A new session could not be created. Details: Appium's IosDriver
does not support xcode version 8.2.1. Apple has deprecated UIAutomation. Use the "XCUITest" automationName capability instead. (WARNING: The
server did not provide any stacktrace information)
Command duration or timeout: 617 milliseconds
Build info: version: '2.53.1', revision: 'a36b8b1cd5757287168e54b817830adce9b0158d', time: '2016-06-30 19:26:09'
System info: host: 'dhcp-india-vpn-vpnpool-10-191-197-146.vpn.oracle.com', ip: '10.191.197.146', os.name: 'Mac OS X', os.arch: 'x86_64',
os.version: '10.12.2', java.version: '1.8.0_101'
Driver info: io.appium.java_client.ios.IOSDriver
	at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
	at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
	at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
	at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
	at org.openqa.selenium.remote.ErrorHandler.createThrowable(ErrorHandler.java:206)
	at org.openqa.selenium.remote.ErrorHandler.throwIfResponseFailed(ErrorHandler.java:158)
	at org.openqa.selenium.remote.RemoteWebDriver.execute(RemoteWebDriver.java:678)
	at io.appium.java_client.DefaultGenericMobileDriver.execute(DefaultGenericMobileDriver.java:42)
	at io.appium.java_client.AppiumDriver.execute(AppiumDriver.java:1)
	at io.appium.java_client.ios.IOSDriver.execute(IOSDriver.java:1)
	at org.openqa.selenium.remote.RemoteWebDriver.startSession(RemoteWebDriver.java:249)
	at org.openqa.selenium.remote.RemoteWebDriver.<init>(RemoteWebDriver.java:131)
	at org.openqa.selenium.remote.RemoteWebDriver.<init>(RemoteWebDriver.java:144)
	at io.appium.java_client.DefaultGenericMobileDriver.<init>(DefaultGenericMobileDriver.java:37)
	at io.appium.java_client.AppiumDriver.<init>(AppiumDriver.java:162)
	at io.appium.java_client.AppiumDriver.<init>(AppiumDriver.java:171)
	at io.appium.java_client.ios.IOSDriver.<init>(IOSDriver.java:56)
	at SampleTestForNativeApplicationInIOS.main(SampleTestForNativeApplicationInIOS.java:26)
```

```
		caps.setCapability(MobileCapabilityType.APP, file);
		caps.setCapability(MobileCapabilityType.PLATFORM_NAME, "iOS");
		caps.setCapability(MobileCapabilityType.PLATFORM_VERSION, "8.4");
		caps.setCapability(MobileCapabilityType.DEVICE_NAME, "iPhone 6");
```

# Automating app on iOS 10+

**Introduction:-**

Appium drives various native automation frameworks and provides an API based on Selenium's WebDriver JSON wire protocol.

For new iOS versions (9.3 and up), Appium drives Apple's XCUITest library. Our support for XCUITest utilizes Facebook's WebDriverAgent project.

For older iOS versions (9.3 and below), Appium drives Apple's UIAutomation library, using a strategy which is based on Dan Cuellar's work on iOS Auto.

Android support uses the UiAutomator framework for newer platforms and Selendroid for older Android platforms.

Windows support uses Microsoft's WinAppDriver

## Automating app on iOS 10+ , continuation..

**Migrating your iOS tests from UIAutomation (iOS 9.3 and below) to XCUITest (iOS 9.3 and up)**

- For iOS 9.2 and below, Apple's only automation technology was called **UIAutomation**, and it ran in the context of a process called "Instruments".
- As of iOS 10, Apple has completely removed the UIAutomation instrument, thus making it impossible for Appium to allow testing in the way it used to.
- Fortunately, Apple introduced a new automation technology, called XCUITest, beginning with iOS 9.3. For iOS 10 and up, this will be the only supported automation framework from Apple.

**List of softwares to be installed:-**

## Reference

- https://developer.android.com/studio/command-line/adb.html

# Latest updates

```
Locator Strategy 'name' is not supported for this session
```

Name locator strategy is deprecated instead use //*[@**name**="Animation"]