

# Mobile Test Automation Documentation Academy

## **Index:**

### **ENVIRONMENT CONFIGURATION FOR MOBILE AUTOMATION**

1. Verify that JDK is installed correctly.
2. Download and install Android SDK
4. Create a virtual device
5. Install Appium Desktop
6. Install the app
7. Opening and inspecting the application
- 8, AppPackage and AppActivity

# ENVIRONMENT CONFIGURATION FOR MOBILE AUTOMATION

## 1. Verify that JDK is installed correctly.

If you need to install JDK, follow these steps:

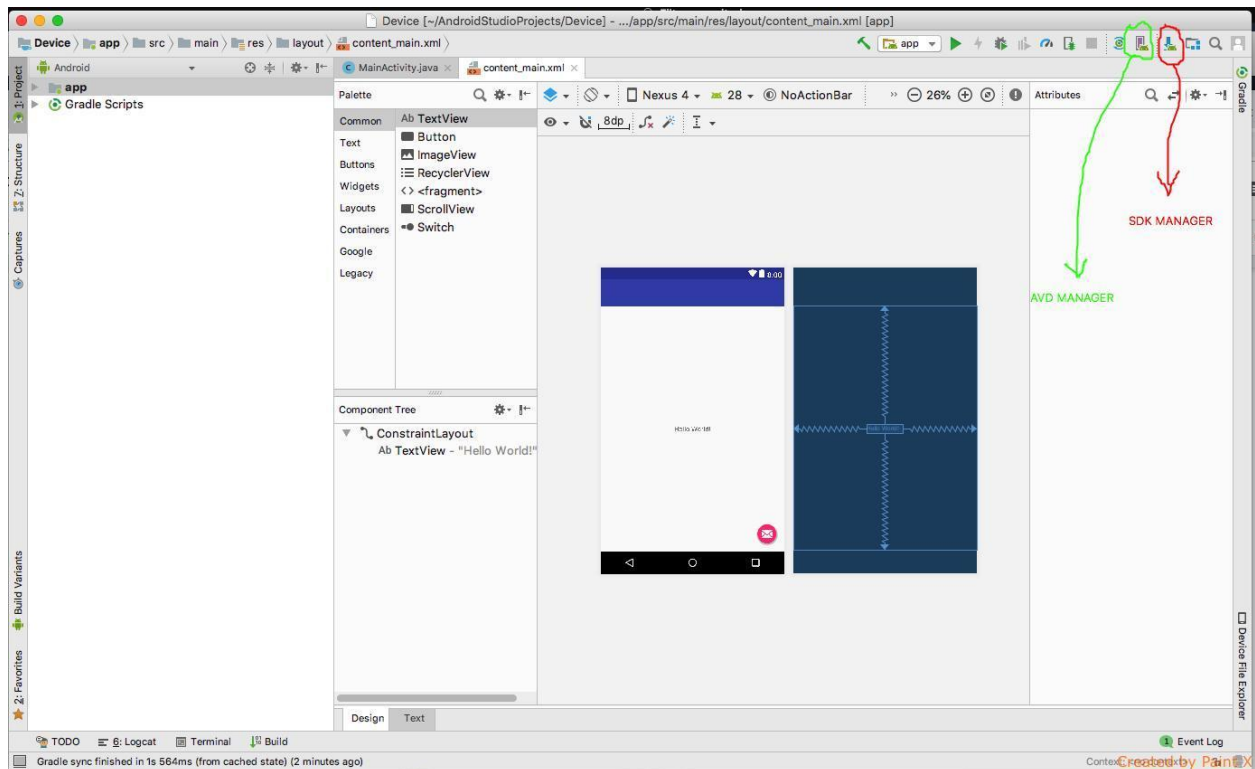
- Download and install JDK 8 from <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>!
- **Optional:** You can determine if the JDK was successfully installed by typing “java -version” in a Terminal window. You will see something like this:

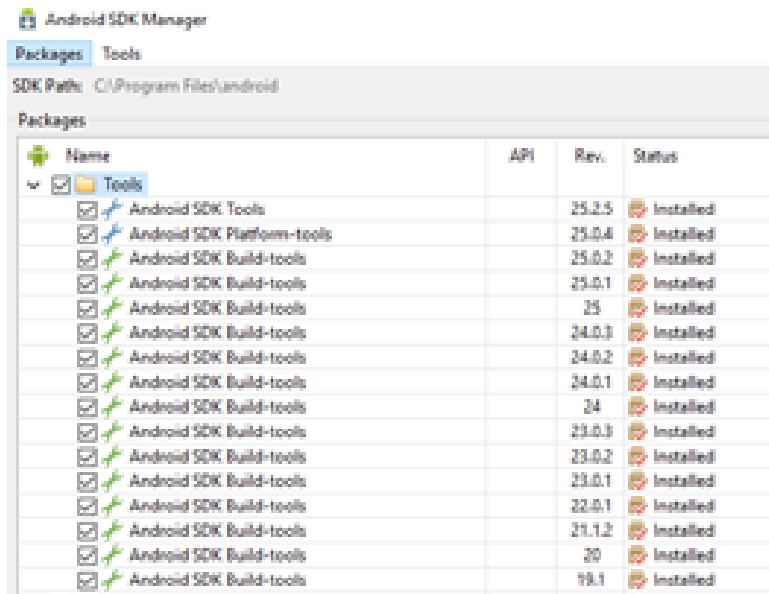
```
% java -version
java version "1.8.0_06-ea"
Java(TM) SE Runtime Environment (build
1.8.0_06-ea-b13)
Java HotSpot(TM) 64-Bit Server VM (build 23.2-b04,
mixed mode)
```

- Set JAVA\_HOME to the path of the jdk (ie. C:\Program Files\Java\jdk1.8.0\_###)

## 2. Download and install Android SDK

- Download and install Android Studio from <https://developer.android.com/studio/?hl=es-419>
- Run SDK MANAGER





Install all tools and extras, include some version of android.

- Set ANDROID\_HOME to the path of the Android (ie. C:\Program Files\Android\android-sdk)

Add the location of this 3 folders to PATH environment, for example:

(C:\Program Files\android\build-tools) --> ;%ANDROID\_HOME%\build-tools;

(C:\Program Files\android\platform-tools) → ;%ANDROID\_HOME%\platform-tools;

(C:\Program Files\android\tools)--> ;%ANDROID\_HOME%\tools;

PATH = ....

;%ANDROID\_HOME%\tools;%ANDROID\_HOME%\platform-tools;%ANDROID\_HOME%\build-tools;

- Verify that SDK is installed correctly:

Write this in command prompt window (adb) You will see something like this:

Android Debug Bridge version 1.0.39

Revision 5943271ace17-android

### 3. Create a virtual device

Use the AVD manager to create a new virtual device, when the device is created, open it and run in a terminal window this command (`adb devices`) to know the ID of the device or the `deviceName` that we will use as a capability.

### 4. Install Appium Desktop

From <https://github.com/appium/appium-desktop/releases/>

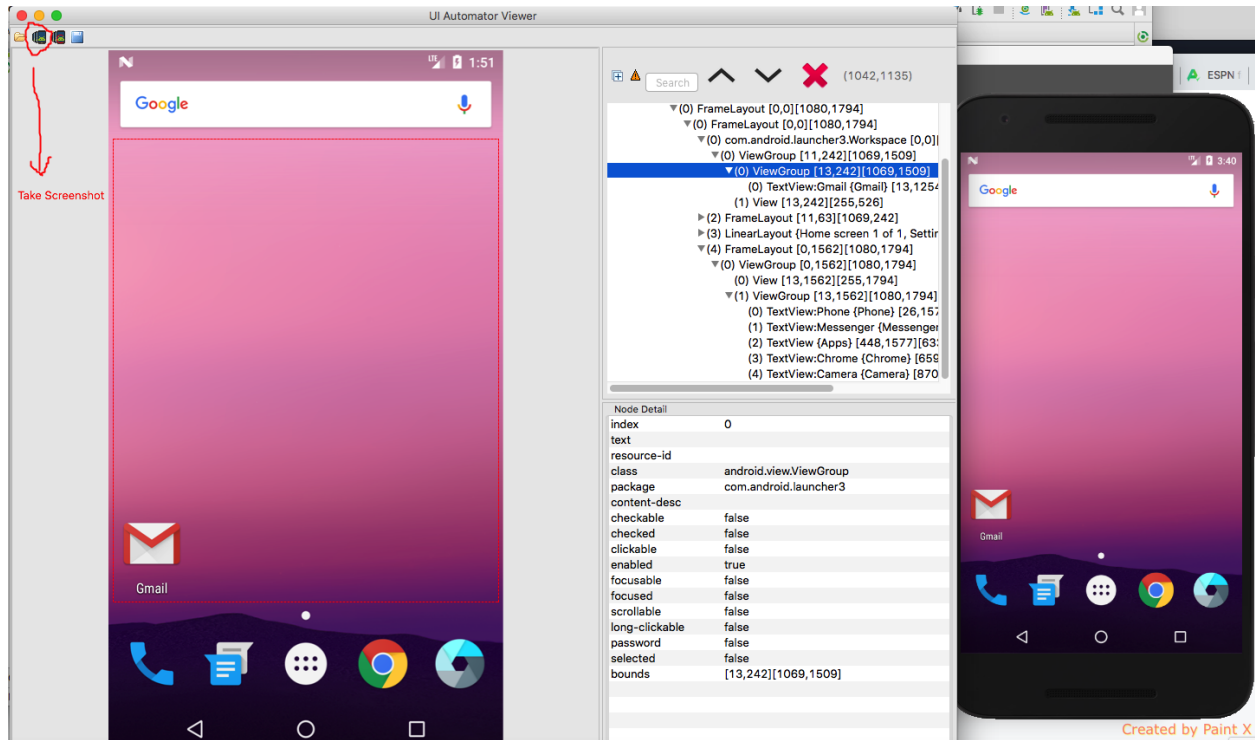
### 5. Install the app

When we have the .apk of an application we can drag it directly to the device or in a terminal window navigate to the folder where the .apk is and run this command (`adb install AppName.apk`)

### 6. Opening and inspecting the application

UiAutomatorViewer:

Open the application in the device and then open the UiAutomatorViewer from your folder `/Android/sdk/tools/bin/UiAutomatorViewer`



## Appium:

Open Appium desktop and start the server, then click on the search button icon and set the next capabilities:

```
{
  "deviceName": "MyDevice",
  "appPackage": "com.MyAppPackage",
  "appActivity": "com.MyAppActivity",
  "platformName": "Android",
  "app": "C:/PathOfMyApplication.apk" (OPTIONAL)
}
```

Automatic Server Custom Server SAUCE LABS TestObject A PART OF SAUCE LABS headspin BrowserStack

Will use currently-running Appium Desktop server at <http://localhost:4723>

> Advanced Settings

Desired Capabilities Saved Capability Sets (4) Attach to Session...

deviceName	text	emulator-5554	
appPackage	text	com.rccl.royalcaribbean	
appActivity	text	com.rccl.excalibur.activ	
platformName	text	Android	
app	text	C:/MyApplicationPath	

JSON Representation

```
{
  "deviceName": "emulator-5554",
  "appPackage": "com.rccl.royalcaribbean.debug",
  "appActivity": "com.rccl.excalibur.activity.SplashActivity",
  "platformName": "Android",
  "app": "C:/MyApplicationPath"
}
```

[Desired Capabilities Documentation](#) Save Save As... Start Session

Start the session.

App Source

Refresh and take the screenshot

Selected Element

Find By Selector

Find By	Selector
accessibility id	Sign in
id	com.rccl.royalcaribbean.debug:id/launch_pad_sign_in
xpath	//android.widget.Button[@content-desc="Sign in"]

Attribute Value

Attribute	Value
index	4
text	Sign in
class	android.widget.Button
package	com.rccl.royalcaribbean.debug
content-desc	Sign in
checkable	false
checked	false

Created by PaintX



## 8. Note: How know the **appPackage** and **appActivity**?

<https://support.testsigma.com/support/solutions/articles/32000019977-how-to-find-app-package-and-app-activity-of-your-android-app>

In a **windows**, terminal write this command with the app opened in the device

adb shell (enter)

dumpsys window windows | grep -E 'mCurrentFocus|mFocusedApp' (enter)

```
co-it01458:~ helberth.bolivar$ adb shell
generic_x86:/ $ dumpsys window windows | grep -E 'mCurrentFocus|mFocusedApp'
mCurrentFocus=Window{ad54744 u0 com.google.android.gm/com.google.android.gm.welcome.WelcomeTourActivity};
mFocusedApp=AppWindowToken{f049202 token=Token{abf174d ActivityRecord{e486e4 u0 com.google.android.gm/.welcome.WelcomeTourActivity t17}}};
generic_x86:/ $
```

Created by Paint X

**For Mac/Linux:**

adb shell dumpsys window | grep -E 'mCurrentFocus'