Appium Documentation

Installation:



## First check/install Java in the machine

Download JDK, run the exe, it will ask for the path where it should install JDK and JRE for both provide the path where it should be installed, Usually it will be in C drive.

Create JAVA\_HOME in environmental variables

JAVA\_HOME --- paste the path till jdk1.09etc

In Path provide path till bin--------- …..\bin

## Download Android studio

Download android studio from the official website

An exe will be downloaded, then u will need to run it.

After the successful completion of the run, in C:\users\Appdata should be created(it might be in invisible mode).

If it is unvisible go to C:\users\ then in the folder path type %AppData% you should see the folder and sub folders in AppData.

## Paths that needs to be set for Android

AppData\local\Android\sdk will be there.

There are 4 variables that needs to be set

Now create Environment variables for Android

Create ANDROID\_HOME and provide path till SDK **\..\..\..\sdk**

Provide under path the below three paths…

**..\Andorid\sdk\tools\bin**

**..\Andorid\sdk\tools**

**..\Andorid\sdk\platform-tools**

## Working with Android studio

We will be using android studio to work with AVD and SDK managers

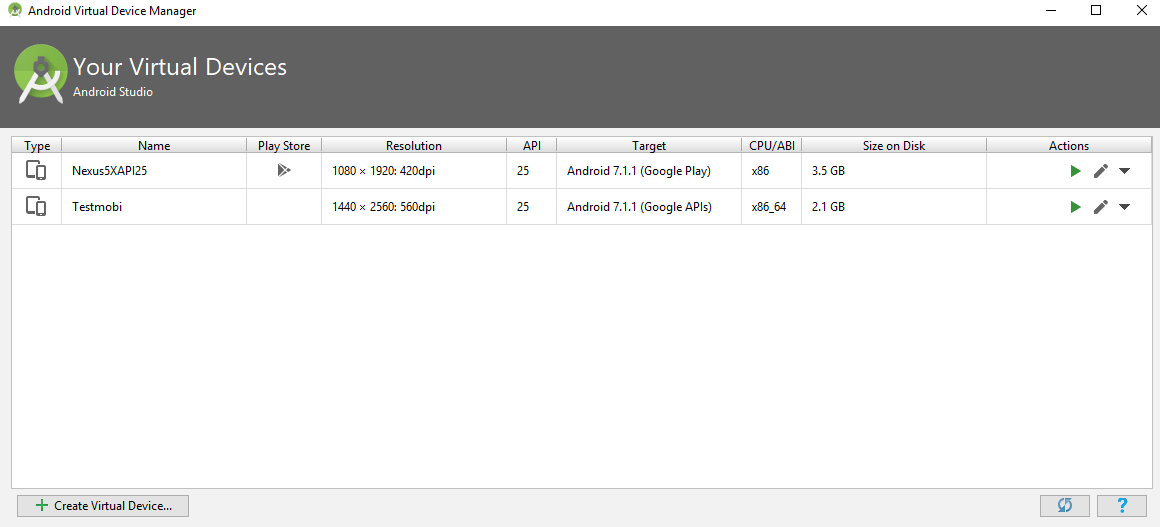
Programfiles\andorid\android studio\bin

Click on studio64

Navigate till sample project

## Configure virtual device

Click on AVD manager icon



You will see a screen as above

Click on create virtual device, rest of the screens will be self-explanatory, after that it will be displayed click on the first icon under Action column.

Sometimes it won’t launch, then you can check the reason in android studio.

***Troubleshooting for device not opening***

Open SDK Manager and Download Intel x86 Emulator Accelerator (HAXM installer) if you haven't.

Now go to your SDK directory (C:\users\username\AppData\Local\Android\sdk, generally). In this directory Go to extra > intel > Hardware\_Accelerated\_Execution\_Manager and run the file named "intelhaxm-android.exe".

In case you get an error like "Intel virtualization technology (vt,vt-x) is not enabled". Go to your BIOS settings and enable Hardware Virtualization.

Restart Android Studio and then try to start the AVD again.

When you switch on your device, it starts to boot as your preferences (HDD, Removable drive, Optical drive etc.) set in the BIOS. Before the OS boot starts, if you press a particular key,your device will show a window with options like Boot menu, BIOS settings etc. You can find the buttons beside each option, by pressing which you can enter those options. For example,

Acer - F2

Asus - Del

DELL - F2

HP - F10

These keys may be changed.

After you enter the BIOS set up goto the Advanced tab and enable the Intel virtualization technology, which may be aliased/called as Intel VT-x or AMD-V

If it is related to HAXM—related to intel then

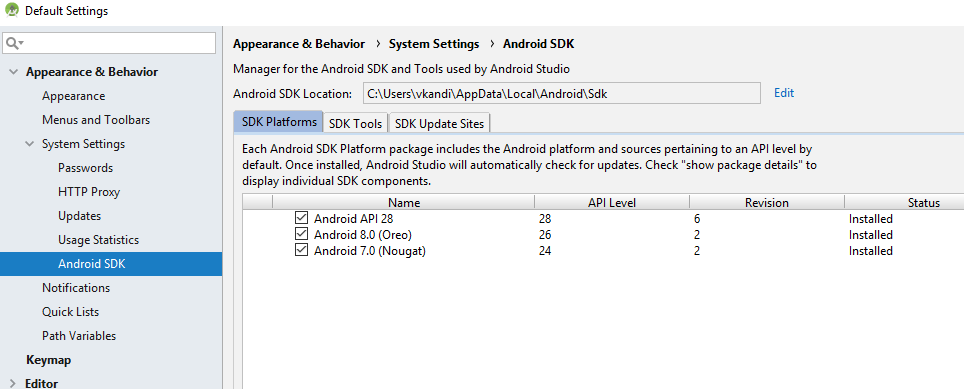
C:\Users\xxxx\AppData\Local\Android\Sdk\extras\intel\Hardware\_Accelerated\_Execution\_Manager

Under the above

haxm\_check.exe and also intelhaxm-android run this and then the virtual device will be opened.

Download the android type while creating the virtual device, select/download the stable android version where it has Appium support.

### Check SDK manager:



Check SDK platforms, SDK tools

Where you can see what are all the things that got installed, if anything else is required you can download from here.

### Open virtual device without android studio

First navigate to the platform-tools

C:\Users\xxx\AppData\Local\Android\Sdk\platform-tools

As Adb presents in that folder.

After navigating to the above mentioned path in cmd

emulator –avd devicename

you can see the device opened.

## Downloading Node.js and install Appium via npm

Appium comes as a Node module.

To work with Appium we need node.js on the machine.

Download node js from the below link

<https://nodejs.org/en/download/>

select the msi file which suits your machine specifications.

After the software gets downloaded, install it and it will be installed in the path, C:\Program Files\nodejs\

After installation you can see nodejs file under program files.

### Setting environment variables for nodejs

**Create a sys system variable with name NODE\_HOME and value as C:\Program Files\nodejs**

Node is a software and NPM is used to download node modules like angular js, protractor etc.

**In Path variable give the path C:\Program Files\nodejs\node\_modules\npm\bin**

## **Appium installation**

### Appium server

We can install Appium in two ways

1. Installing using npm
2. Installing using GUI

#### Installing using NPM

Appium client is like java/C/python or any other supported language client

Appium server is the one which listens to the client and then interact with the mobile devices based on the client input.

It is mandatory to invoke server before starting exection.

We need to download client and server separately in our machine.

It is a node module, so it should be downloaded by npm(command line installer, which helps to install node modules)

*> npm install –g appium*

g in the above means globally in the system.

If you are in office network then there will be issue in installing appium because of proxy settings so those needs to be addressed to overcome proxy.

After installation completed type appium in the command prompt

*>appium*

You have to see message like welcome to appium…

Here if you are installing in some restricted environment you need to check the proxy and proceed.

#### Installing using GUI

Go to <http://appium.io/downloads.html>

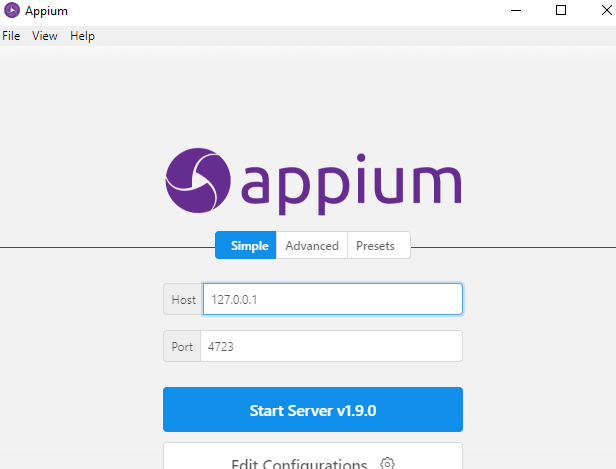
Click on [Appium-Desktop for OSX, Windows and Linux](https://github.com/appium/appium-desktop/releases/latest)

Download the below

**appium-desktop-setup-1.7.0.exe** for windows.

Run the exe for the specific user.

After the installation gets completed. Open Appium from programs and then provide the below info



Host: 127.0.0.1

Port : 4723

Click on start server , so that the server will be started.

### Appium Client

Go to <http://appium.io/downloads.html>

Under Appium Client Libraries select the client in which you are writing code

Here it is java

Download the jar and keep it in the build path of the project that you are working along with selenium jars.

## Basic Program on Appium

Here in the below program I am connecting to the adb device and opening gallery app in that mobile.

To work with a mobile we need to know the configurations of that mobile/virtual device

***First know what are the devices that are available using the command***

adb devices

C:\Users\xxxx>adb devices

List of devices attached

emulator-5554 device

***next check the device details using the below***

C:\Users\vkandi>adb devices -l

List of devices attached

emulator-5554 device product:sdk\_google\_phone\_x86\_64 model:Android\_SDK\_built\_for\_x86\_64 device:generic\_x86\_64 transport\_id:1

now you need to get appPackage and appActivity for the application that you want to interact in the mobile.

For this there are two ways

1)install apkinfo app from playstore, which is used to get the app related info of any application that is installed in the mobile

2)from the command prompt

http://www.automationtestinghub.com/apppackage-and-appactivity-name/

WebDriver driver;

DesiredCapabilities capabilities = **new** DesiredCapabilities();

capabilities.setCapability("BROWSER\_NAME", "Android");

//capabilities.setCapability("VERSION", "4.4.2");

capabilities.setCapability("deviceName","emulator-5554");

capabilities.setCapability("platformName","Android");

// This package name of your app (you can get it from apk info app)

// capabilities.setCapability("appPackage", "com.android.calculator2");

capabilities.setCapability("appPackage", "com.android.gallery3d");

// This is Launcher activity of your app (you can get it from apk info app)

capabilities.setCapability("appActivity","com.android.gallery3d.app.GalleryActivity");

//Create RemoteWebDriver instance and connect to the Appium server

//It will launch the Calculator App in Android Device using the configurations

//specified in Desired Capabilities

driver = **new** RemoteWebDriver(**new** URL("http://127.0.0.1:4723/wd/hub"), capabilities);

System.***out***.println("Testing done");

## SAMPLE App’s to Test

**API Demos**

## Remote-debugging/connecting to device

**https://developers.google.com/web/tools/chrome-devtools/remote-debugging/**

## Gestures

For any mobile gesture we need to use TouchAction

### Tap & Longpress

TouchActions t = new TouchActions(driver);



In the above we need to import packages manually.

For all the packages that are used we can check in <https://github.com/appium/java-client>.

### Swipe



Right view continuation--->



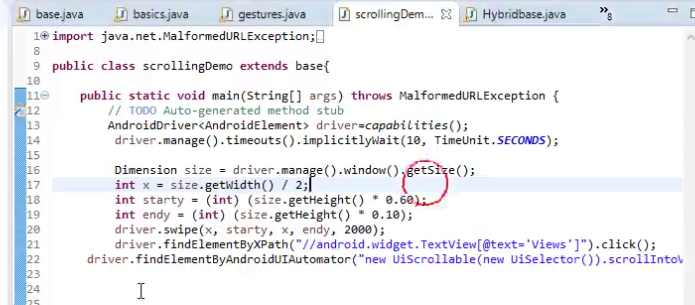
### Scroll

The below scroll code is Android API code and it is not appium code.





But on real device we need to add code based on the screen size



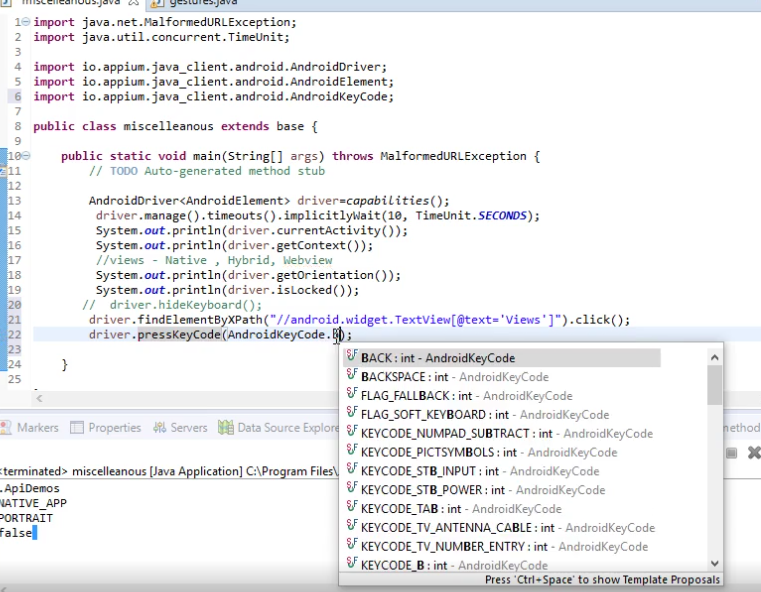
### Drag&Drop



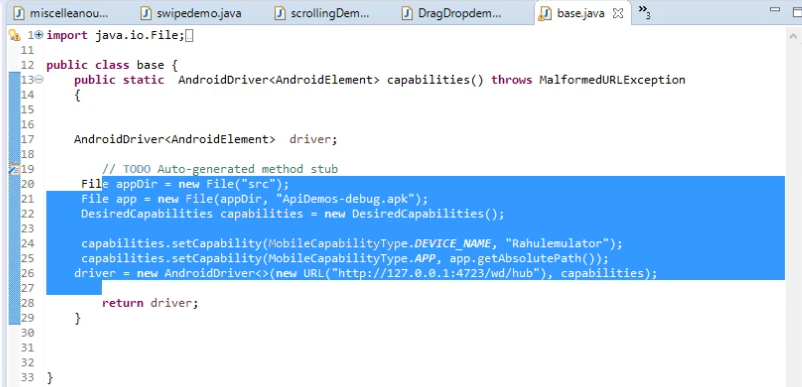
### TAP on Toggle with co-ordinates



### Misc



## Desired capabilities for virtual device



## Invoking App with Package Activity

Without knowing .apk file we can invoke an app using package and activity name of that App.

Steps:

Download apk info app

There you can view the app for which you want to install



## Device Connections:

### Stop and start the server

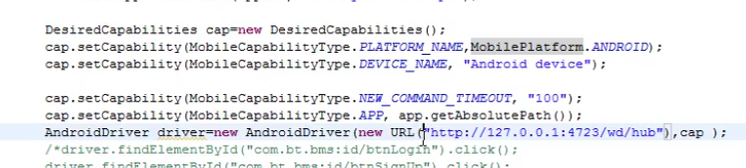
If device is not shown as connected then use the below mentioned adb command



adb kill-server

adb start-server

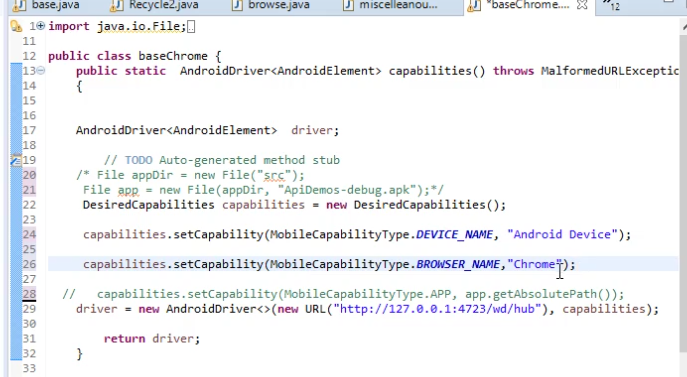
### To run on real-device



DEVICE\_NAME should be Android device for real device

Android emulator for virtual device.

### Invoking chrome browser



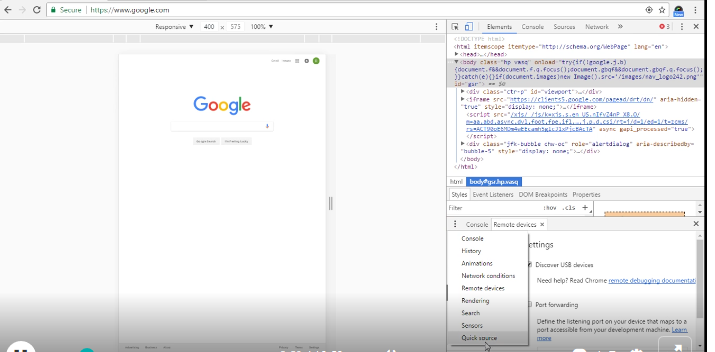
## Identifying objects in mobile browser

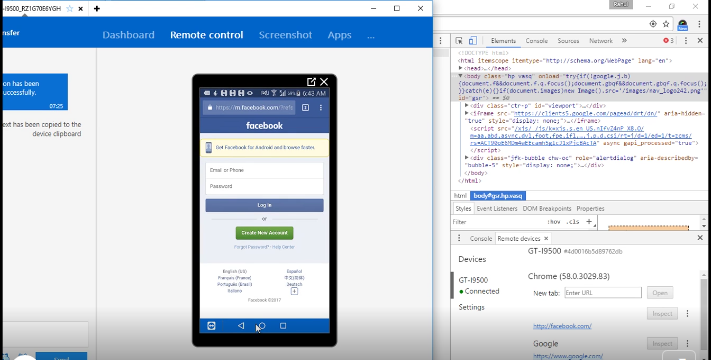
Open chrome browser in the machine

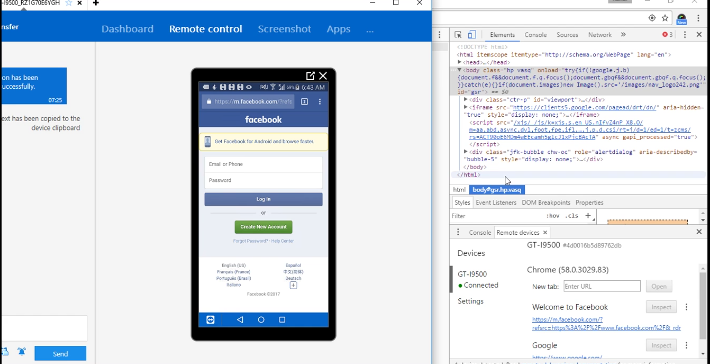
Check whether the device is connected to the machine using adb devices command.

If connected then,

Settings—more tools—developer tools—select remote devices from the hamburger menu option







Now click on the link below welcome to facebook, it opens mobile view on the desktop browser

Now we can get all the objects easily and it works on the mobile

## Scroll in mobile chrome browser



# Hybrid Apps



Apps that have a native as well as web involved in it is know as hybrid.

For example zerodha app is a native app and when u click on help it leads to web browser from app, this is a kind of hybrid app.

As in the above code snippet , driver.getContext() gives the current context i.e, native or web

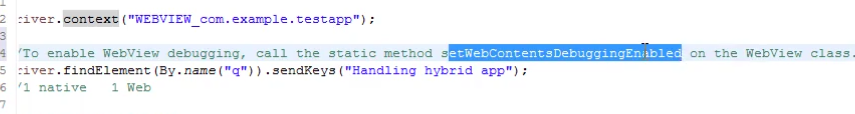
driver.getContextHandles() gives the available types i.e, native and web

We need to switch the context then only we can handle web from native and viceversa. So we need to give the below command

driver.context(“context name”)

**sometimes you might not see the web context through driver.getContextHandles()** then you need to request developer on the below

you need to ask for webview debugging, they need to call the static method setWebContentsDebuggingEnables on the WebView class



## Exception

A class name cannot be taken as a tag name in xpath if it has special characters like $