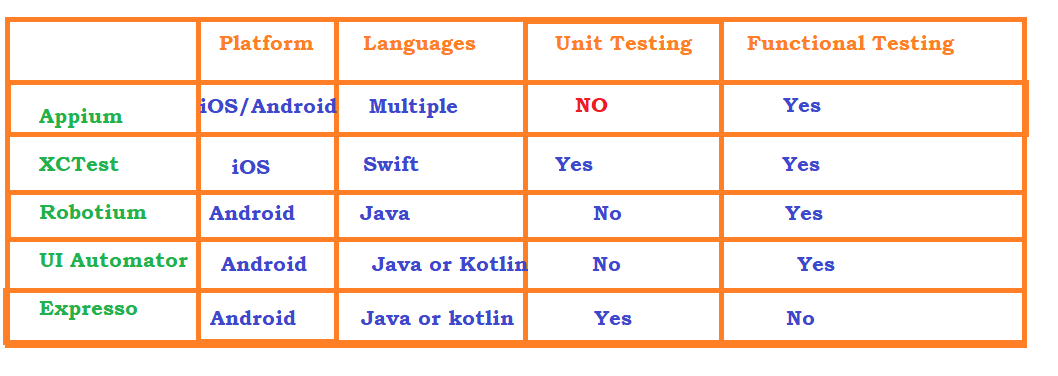
Mobile Automation with Appium

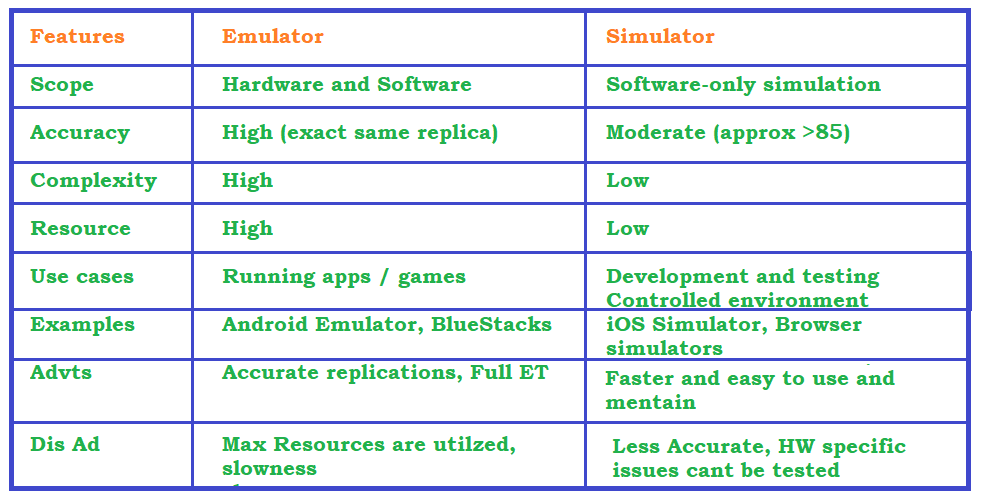
# Features :

* Open source freely available software
* Appium supports both Android and iOS
* Any kind of mobile applications can be automated
  + Native apps
  + Hybrid apps
  + Web apps
* Supports multiple programming language
  + Java
  + python
  + Ruby
  + C#
  + php
  + java script
* Internally Appium uses selenium WebDriver APIs
* Appium Supports physical device as well as android emulator or an iOS Simulator
* Don’t have to re-compile or modify the automation code for iOS and Android
* Appium is developed and supported by sauce labs and browser stack
* Good support community

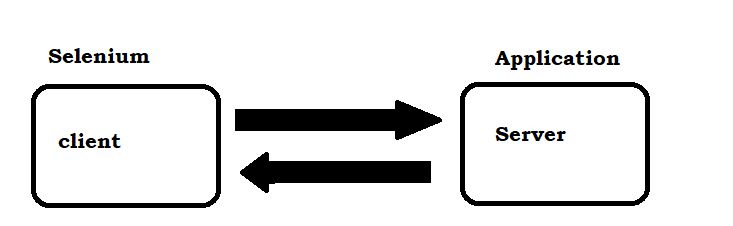
# Comparing tools :



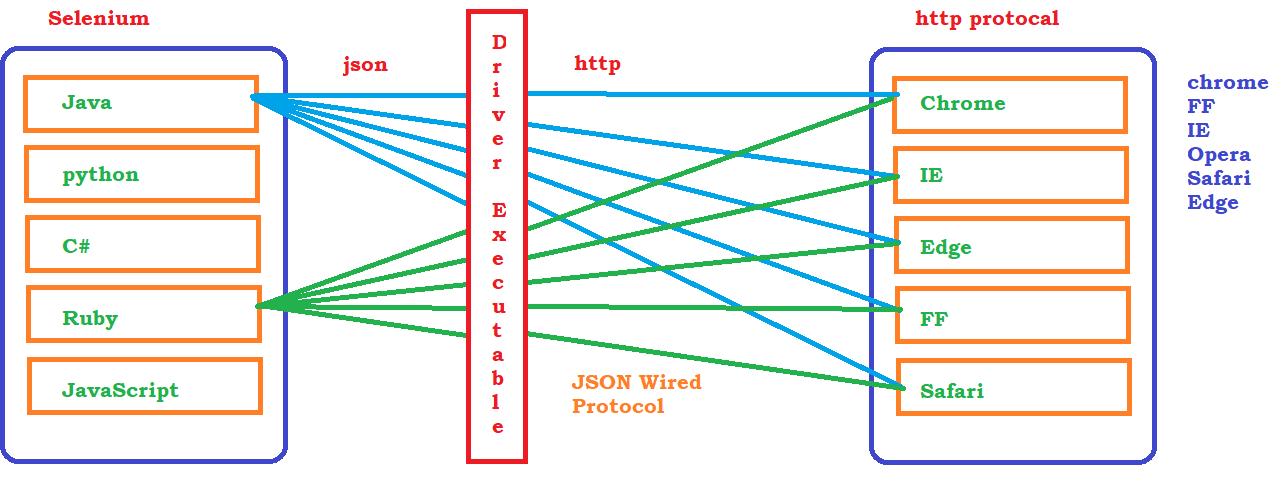
# Difference Between Emulator and Simulator

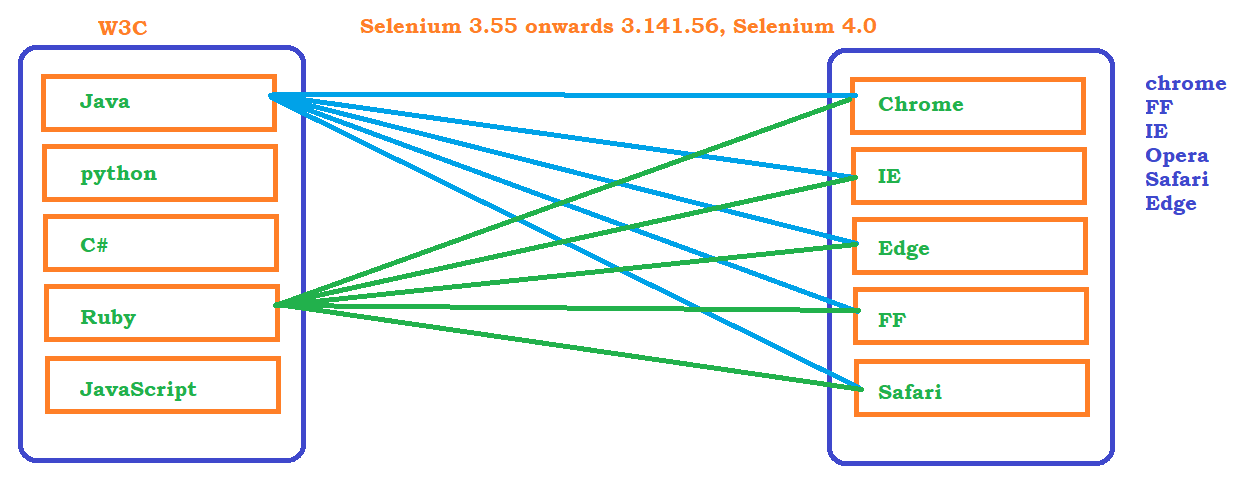


# Appium Architecture

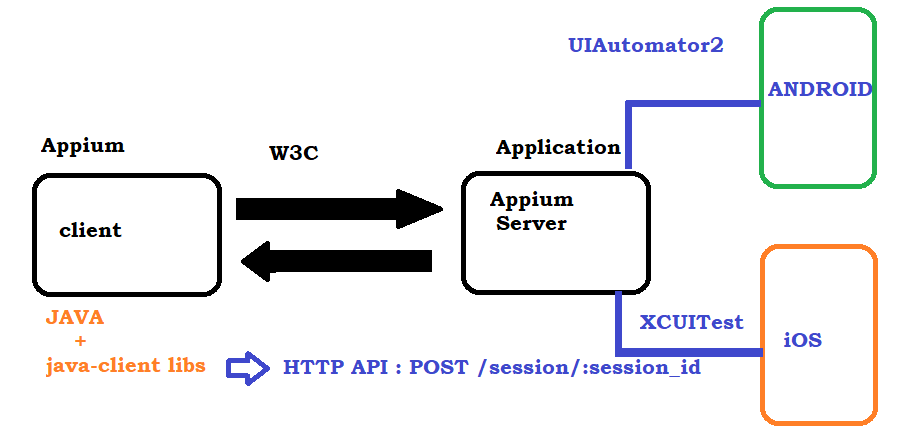


# Json Wired protocol





# Drivers for Mobile Apps

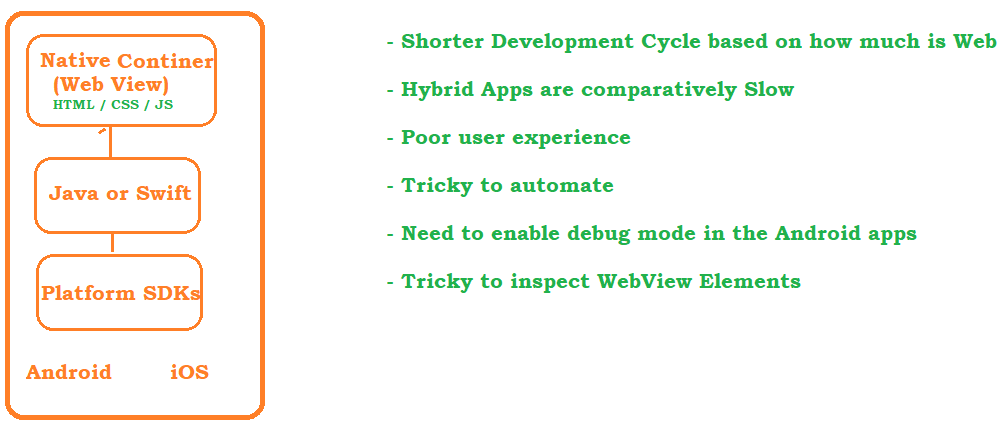


# Different types of Mobile Application

1. Native Apps :



1. Hybrid Apps :



1. Web App / Mobile Web

These are the applications which are opened on a mobile browser. even though there is a native app, some applications support separate web browser implementation for Web page and mobile browser

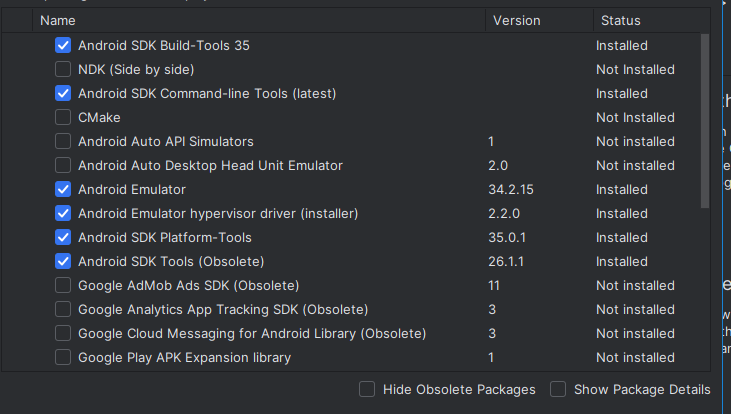
Appium Setup

# Download and Install JDK 8 / JDK-17

* + Download and install jdk-17 from the official website <https://download.oracle.com/java/17/archive/jdk-17.0.10_windows-x64_bin.exe>
  + JAVA\_HOME : C:\Program Files\Java\jdk-17
  + PATH : C:\Program Files\Java\jdk-17\bin

# Download and install Android Studio : it comes with 2 packages

* Download and install from below link <https://developer.android.com/studio>
  1. Android SDK : Comes with all the required software to run emulator
  2. Android Studio : Creating a Virtual Device
* Make sure all the drivers are installed successfully, if there is a failure then follow the debug steps to install the missing softwares
* Create a New Virtual Device :
  1. Open Android SDK and create a dummy project with **Empty Activity** option
  2. Once the build is complte Verify the SDK Tools
* Verify SDK Tools
  1. click on Settings and open SDK Manager



* NOTE : SDK by default will be installed insdie

C:\Users\Aravind\AppData\Local\Android\Sdk

* Set Environment Variables:
  1. ANDROID\_HOME (SDK Location) : *C:\Users\Aravind\AppData\Local\Android\Sdk*
  2. Update the PATH :
     + C:\Users\Aravind\AppData\Local\Android\Sdk\tools
     + C:\Users\Aravind\AppData\Local\Android\Sdk\tools\bin
     + C:\Users\Aravind\AppData\Local\Android\Sdk\platform-tools
  3. under platform-tools we have adb.bat( Android Debug Bridge ) file which is a command line **tool** that allows us to communicate with the android device. Using adb command we can install, uninstall software and list the devices connected to the machine
  4. **uiautomatorviewer :** command line tool to inspect the Mobile Elements
  5. **emulator :** used to launch the emulator

# Create a new Virtual Device

* Go to Android Studio
* Click on Tools -> Device Manager
* Click on + icon followed by Create Virtual Device
* Select the required phone from the list displayed and click on next
* Download or select the required OS Version from the menu
* provide AVD Name -> Pixel
* click on `Show Advanced Settings` to provide additional information like network configuration, RAM, Storage, Camera access etc
* click on finish 🡪 you will be getting a virtual phone
* click on Run button to start the phone

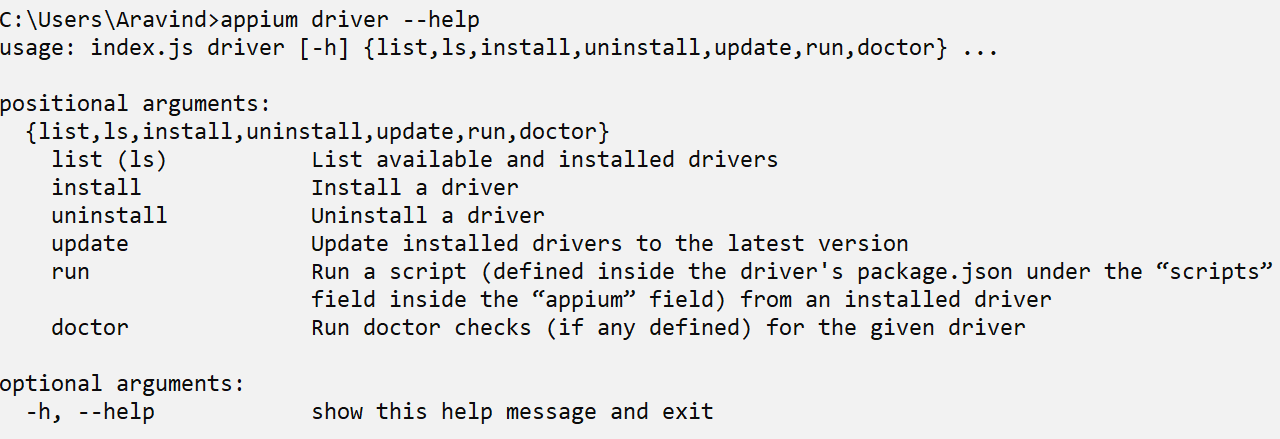
# Install Appium Server

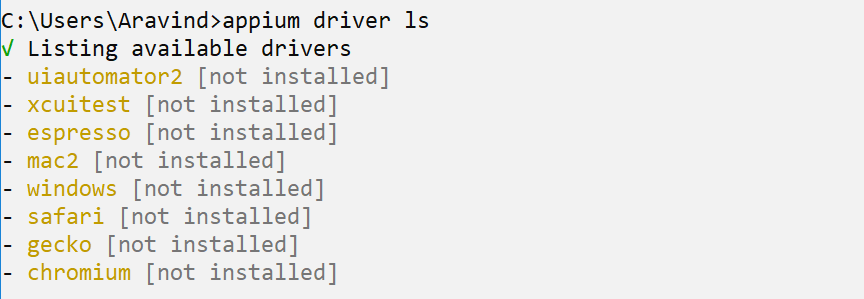
* Install using node software from below link
* <https://nodejs.org/en/download/package-manager>
* confirm the installation is successful by executing
  + node –v
  + npm –v
* Navigate to <https://www.npmjs.com/>
* execute npm –g appium command to install the latest version
* verify appium –version command in the command prompt to verify the successful installation (in case if you get not recognized error then follow the below setps)
  + go to C:\Users\<<id>> \AppData\Roaming\npm location
  + verify appium cmd file is present
  + then update the above path
  + open new command prompt and retest appium –v

# Enable USB Debugging option on Real Device

* go to settings on the real device
* click on more settings -> about phone -> on version tap 7+ times
* you will get a notification on developer mode is enabled
* NOTE : if usb is not detected follow the steps
  + <https://developer.android.com/studio/run/oem-usb>
  + <https://developer.android.com/studio/run/oem-usb#Drivers>

# Installing required Drivers

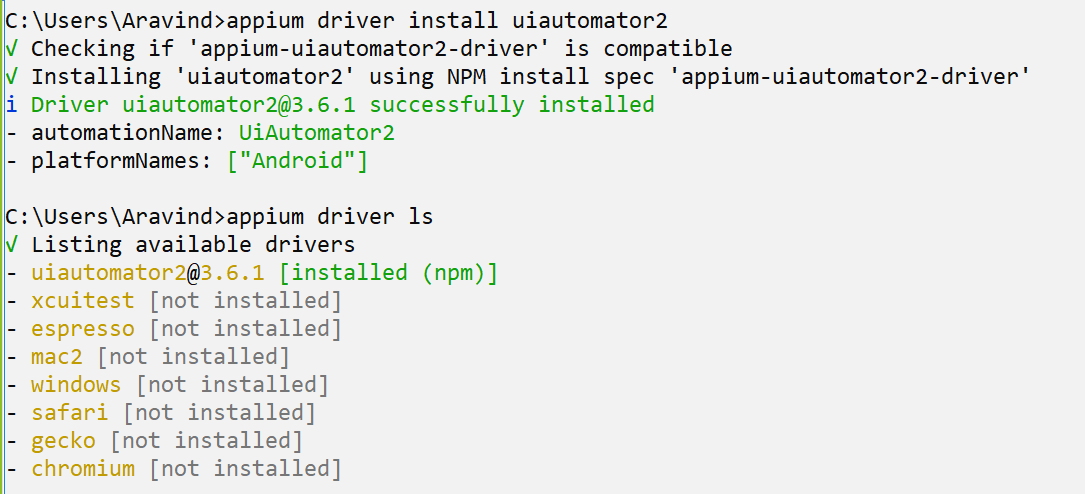
* To know the list of supported drivers, Open command prompt
* **appium driver –help**
* 
* **list of supported drivers**



* **Install the required driver**

**NOTE : Understanding 2.0 Changes**

[**https://appium.io/docs/en/2.0/guides/migrating-1-to-2/**](https://appium.io/docs/en/2.0/guides/migrating-1-to-2/)



# Install Appium Inspector :

* download the application : https://github.com/appium/appium/blob/master/packages/appium/sample-code/apps/ApiDemos-debug.apk

# Create a project and configure Appium

* 1. Open Eclipse
  2. Create a Maven Project
  3. Select groupid : org.apache.maven.archetypes and artifactid : maven-archetype-quickstart and version :
  4. click on finsigh
  5. Update JDK to the latest : Right click on project -> properties -> Javacompiler -> update the compiler to the latest
  6. Update JRE to the latest : Right click on project -> properties -> Java build path -> library -> add library -> select jdk (jre) location
  7. Add Testng Plugin -> Help -> Eclipse market place -> searach for testing and install the plugin
  8. Update pom.xml
     + <https://mvnrepository.com/artifact/io.appium/java-client>
     + <https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java>
     + <https://mvnrepository.com/artifact/org.testng/testng>

NOTE : Migration from 1.0 to 2.0

<https://appium.io/docs/en/2.0/guides/migrating-1-to-2/>

Basic Operations on Mobile App:

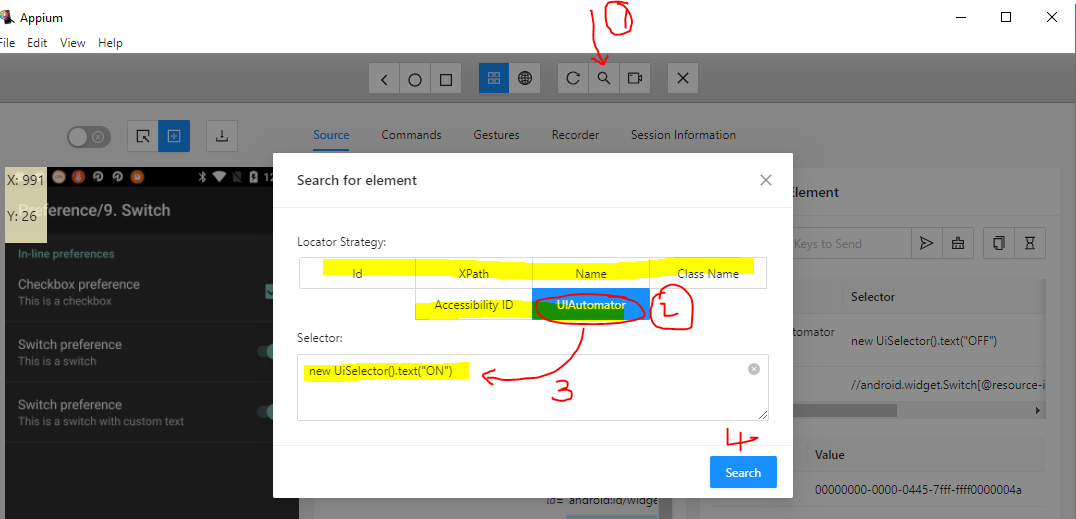
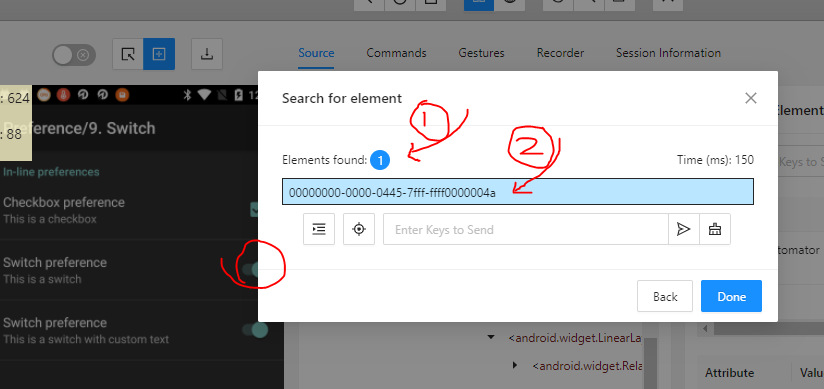
All the identification mechanism which are supported by selenium are also supported by Appium, along with that we have some additional identifiers exclusively for Appium

* id
* name
* className
* tagName
* linkText
* partialLinkText
* cssSelector
* xpath

Appium Specific Identifiers :

* accessibilityID(String)
* androidDataMatcher(String)
* androidUIAutomator(String)
* androidUIMatcher(String)
* androidViewTag(String)

# Finding Elements and performing operations in Android Inspector

* Open Android inspector
* Open the application
* click on search button on the top right corner
* select the identify mechanism, write the expression and click on search
* 
* after search
* 
* Perform the operation like click or type using the options available in above screen shot

Xpath writing Techniques :

* Basic xpath
* Logical Operators
  1. and : //th[@class='infobox-label' and text()='Directed by']
  2. or : //th[@class='infobox-label' or text()='Directed by']
  3. not : //th[@class='infobox-label' and not(text()='Directed by')]
* Functions
  1. text() : //th[text()='Directed by']
  2. contains(arg1. arg2)
     + arg1 – can be any attribute or text function
     + arg2 – corresponding value
  3. starts-with(arg1,arg2)
     + arg1 – can be any attribute or text function
     + arg2 – corresponding value
* Traverse from parent to Child : //tr[th[text()='Directed by']]//a
* Traverse from child to parent : //tr[th[text()='Directed by']]//a
* Axes Functions
  1. traverse to following-sibling : //th[text()='Directed by']/following-sibling::td/a
  2. traverse to preceding-sibling
  3. following
  4. preceding
  5. child
  6. parent
  7. ancestor : //span[contains(text(),'Gevora Hotel')]/ancestor::div[@class='flexOne makeFlex']//p[@id='hlistpg\_hotel\_shown\_price']

Handling Dynamic Elements in Appium OR Dependent and independent Elements in Appium

xpath

Advanced mobile gestures

* Scroll
* Swipe
* Drag

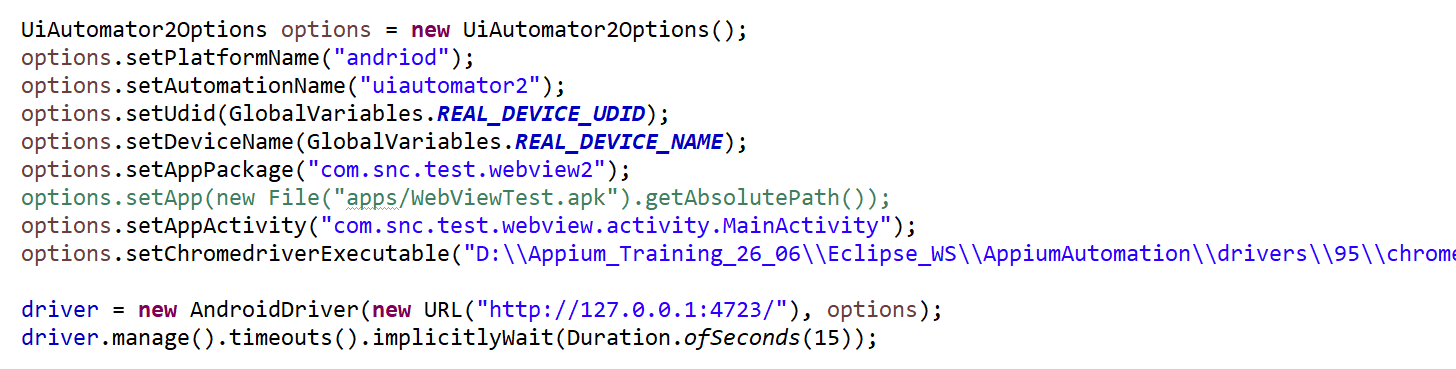
Miscellaneous Functions

ss

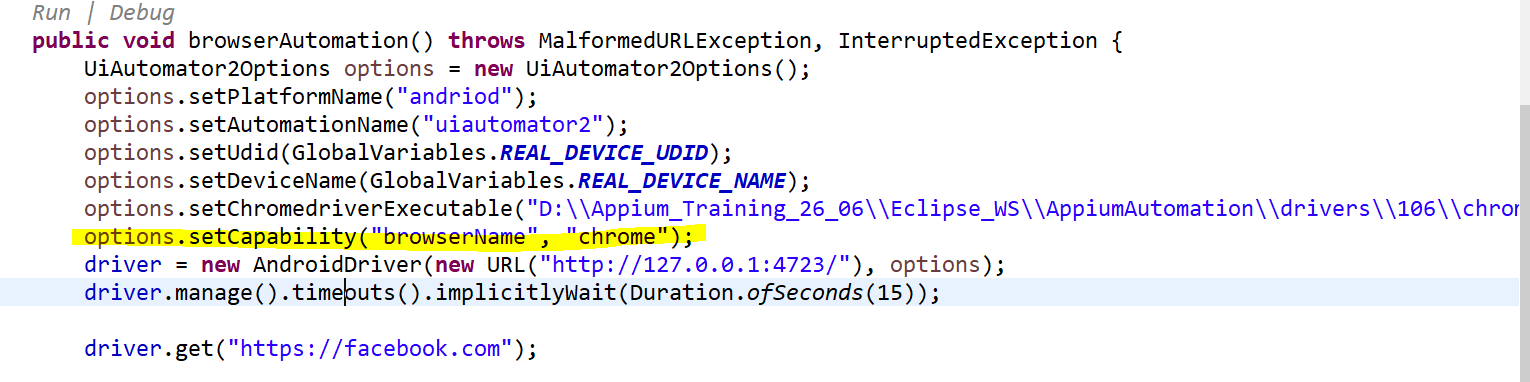
Automating hybrid app

download the required chrome executable from <https://chromedriver.storage.googleapis.com/index.html>

To find the mobile browser version or the driver executable needed, open the browser of hybrid app on mobile and just open browser on the machine and execute **chrome://inspect/#devices**



Automating mobile browser



Framework :

* TestNG
* BDD
* POM