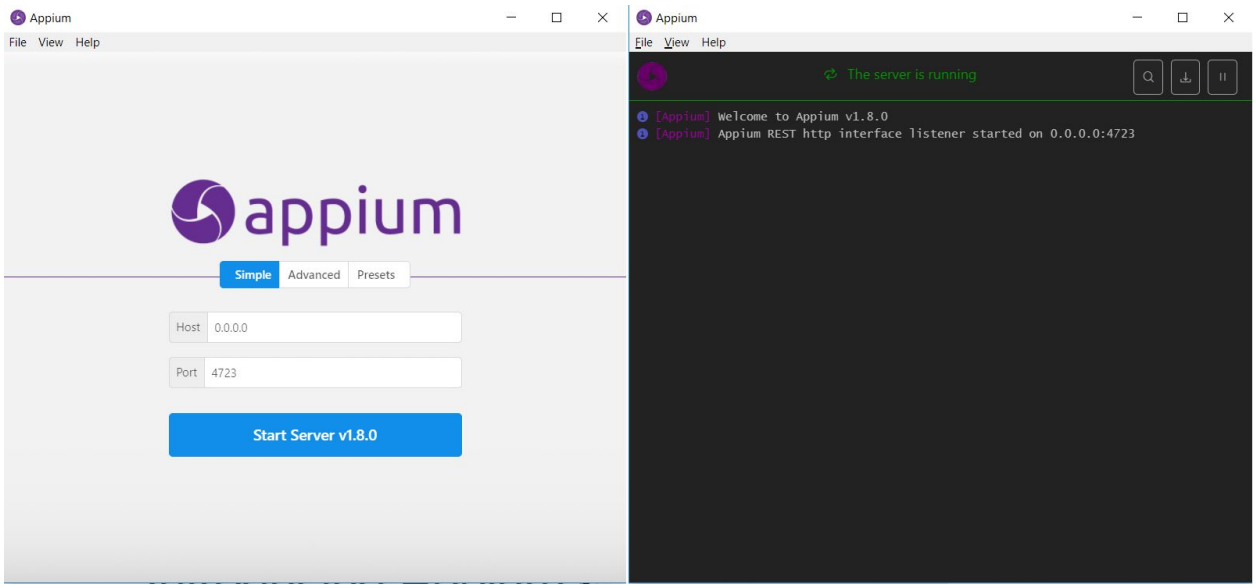
Appium new Series 2018/19:

Appium Tutorials can be found on www.way2automation.com



Initial Android installation:

1. *JAVA_HOME* configured 1.8

2. *Download ANDROIDSTUDIO*

https://developer.android.com/studio/

3. Environment variable setup for *ANDROID_HOME*
    a. VARIABLE NAME: ANDROID_HOME
    b. VARIABLE VALUE: C:\Users\Selenium\AppData\Local\Android\sdk

```
PATH: %ANDROID_HOME%\tools;%ANDROID_HOME%\platform-tools
```

    c. Open SDK Manager
    d. Inside AVD Manager and create a new virtual device.

CONFIGURING ANDROID DEVICE

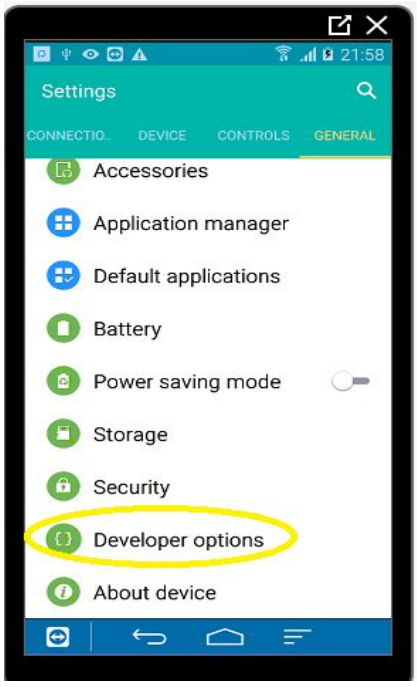
Connect the Device to a Genuine USB cable and make sure drivers are installed completely on your Desktop / Laptop

In some cases if the drivers are not fully installed follow these troubleshooting steps:

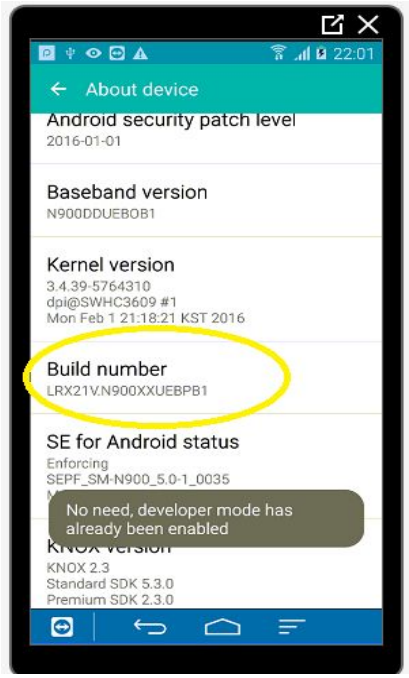
http://theunlockr.com/2009/10/06/how-to-set-up-adb-usb-drivers-for-android-devices/

Once the drivers are fully installed and your machine is able to detect your device

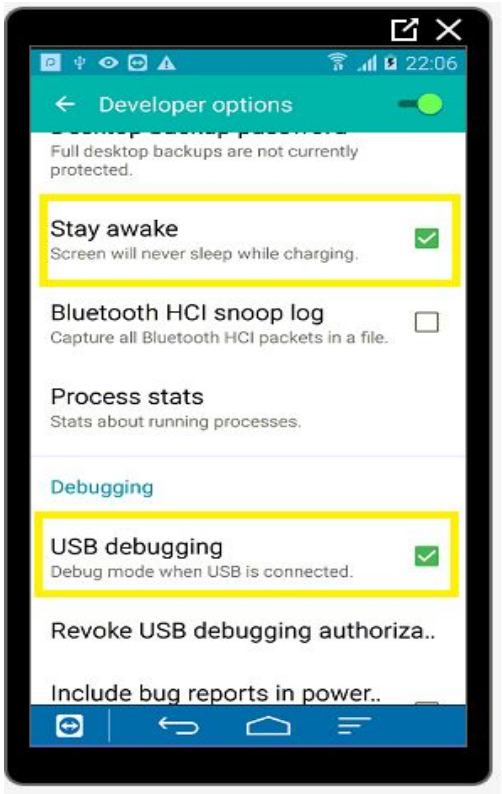
Go to Settings -> General and look for the option “Developer Options”



If you don’t see the Developer Option go to About Device and tap 6 times on Build Number



Go to Developer Options and check USB Debugging and Stay awake options (Make sure to turn off “Stay awake” once done with the testing



Go to Command Prompt and type “adb devices” and make sure you see your device id

Installing Emulator – Genymotion

Emulators are very slow on windows. To get best of the experience with emulators download and install Genymotion Emulator

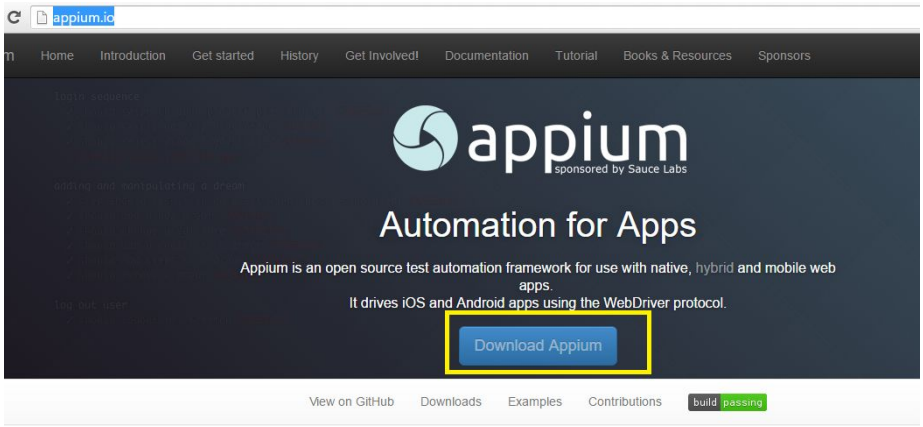
https://www.genymotion.com/features/

DOWNLOAD APPIUM

Appium comes in 2 parts

1. **GUI Tool**

http://appium.io/



2. **Java client (API)**

https://search.maven.org/#search%7Cga%7C1%7Cg%3Aio.appium%20a%3Ajava-client

Maven Dependency

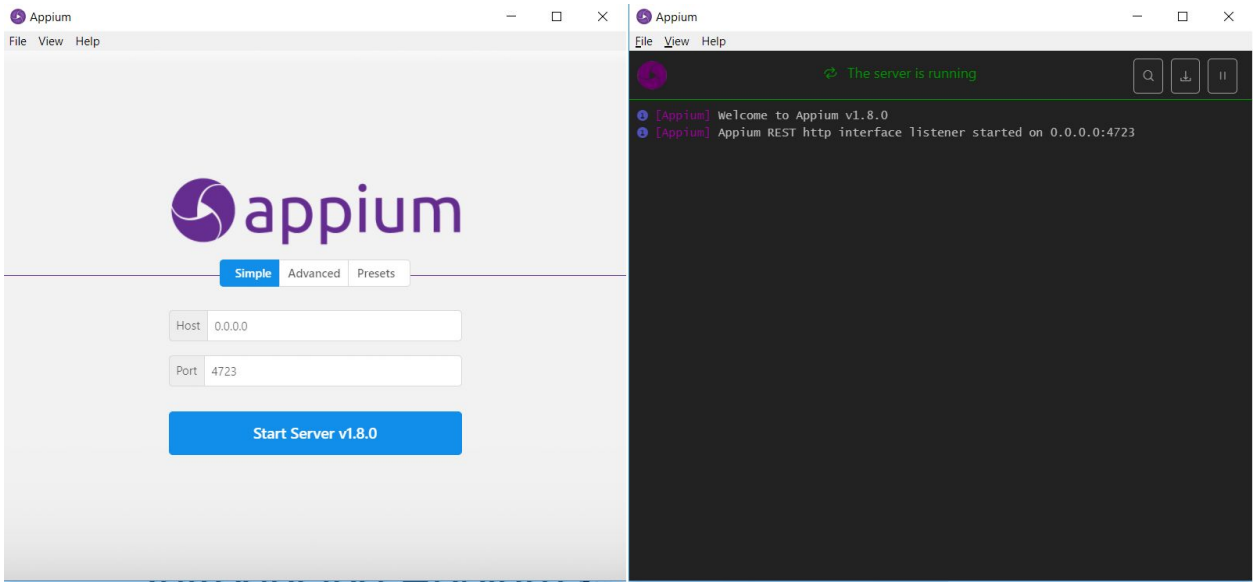
```
<dependencies>


    <dependency>
            <groupId>io.appium</groupId>
            <artifactId>java-client</artifactId>
            <version>6.1.0</version>
        </dependency>

  </dependencies>
```

JAVA Document path

http://appium.github.io/java-client/

STARTING APPIUM SERVER

1: Through Appium App:



2: Through Command Prompt:

Install Nodejs

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

Open terminal and type:

npm install -g appium

MAC OSX INSTALLATION

Java setup and environment variables configured

touch .bash_profile

open .bash_profile

export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_191.jdk/Contents/Home

export PATH=$PATH:$JAVA_HOME

Path for Android Studio on MAC

/Users/rahularora/Library/Android/sdk/

Path for Chromedriver executable in Appium Desktop APP

/Applications/Appium.app/Contents/Resources/app/node_modules/appium/node_modules/

Path for Chromedriver executable in Appium Node

/usr/local/lib/node_modules/appium/node_modules/appium-chromedriver/chromedriver/mac/

Running Appium server Programatically:

WINDOWS:

```
public class TestWebBrowser {

	//AppiumDriver driver = new IOSDriver();

	public static AndroidDriver driver;


	public static void main(String[] args) throws MalformedURLException {

		AppiumDriverLocalService service = AppiumDriverLocalService.buildService(

				new AppiumServiceBuilder().usingDriverExecutable(new File("C:\\Program Files\\nodejs\\node.exe"))

				.withAppiumJS(new File(

"C:\\Users\\WAY2AUTOMATION\\AppData\\Local\\Programs\\Appium\\resources\\app\\node_modules\\appium\\build\\lib\\main.js"))

				.withArgument(GeneralServerFlag.LOCAL_TIMEZONE)
```

```
                                .withLogFile(new File(System.getProperty("user.dir") +
"\\src\\test\\resources\\logs\\log.txt")));



                                service.start();



                DesiredCapabilities capabilities = new DesiredCapabilities();

                //Browser + Device
                capabilities.setCapability(CapabilityType.BROWSER_NAME, "Chrome");
                capabilities.setCapability(MobileCapabilityType.DEVICE_NAME, "Android");



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

                driver.get("http://google.com");
                driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

                driver.findElement(By.name("q")).sendKeys("Hello Appium !!!");
driver.findElement(By.xpath("//*[@id=\"tsf\"]/div[2]/div[1]/div[1]/button")).click();
```

```
                driver.quit();

                service.stop();


        }

}
p on
```

MAC

```
public class TestWebBrowser {

public static AndroidDriver driver;
public static AppiumDriverLocalService service;

        public static void main(String[] args) throws MalformedURLException,
InterruptedException {

                service = AppiumDriverLocalService.buildService(
                                        new AppiumServiceBuilder().usingDriverExecutable(new
File("/usr/local/bin/node"))
```

```
                                                .withAppiumJS(new
File("/Applications/Appium.app/Contents/Resources/app/node_modules/appium/build/lib/
main.js"))

.usingPort(4723).withIPAddress("127.0.0.1")

.withArgument(GeneralServerFlag.LOCAL_TIMEZONE)
                                                .withLogFile(new
File(System.getProperty("user.dir")+"/src/test/resources/logs/Appium.log")));


                service.start();



        DesiredCapabilities capabilities = new DesiredCapabilities();

        //Browser + Device
        capabilities.setCapability(CapabilityType.BROWSER_NAME, "Chrome");
        capabilities.setCapability(MobileCapabilityType.DEVICE_NAME, "Android");



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


        driver.get("http://google.com");
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

```
        driver.findElement(By.name("q")).sendKeys("Hello Appium !!!");

driver.findElement(By.xpath("//*[@id=\"tsf\"]/div[2]/div[1]/div[1]/button")).click();


        driver.quit();

        service.stop();


    }

}
```

(Troubleshoot while Setting Path in the Eclipse : MAC machine same for Windows)

Launch your eclipse ->Click on Run -> Click on Run configurations

Select the Environment Tab -> Now set Android_Home path in the eclipse .

Download Appium GUI + Command Line

#Homebrew

/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

brew install node

npm install -g appium-doctor

run appium-doctor //will only work on mac

/Users/rahularora/Library/Android/sdk

npm install -g appium
npm install wd
brew install carthage

```
public class FirstTest {

    public static AndroidDriver driver;

    public static void main(String[] args) throws
MalformedURLException {
```

```
        DesiredCapabilities cap = new DesiredCapabilities();
        cap.setCapability(CapabilityType.BROWSER_NAME,
"Chrome");
        //make sure to download the chrome browser as per version
        //cap.setCapability("chromedriverExecutableDir",
"/usr/local/lib/node_modules/appium/node_modules/appium-chromedriver/c
hromedriver/mac/");


        cap.setCapability(MobileCapabilityType.DEVICE_NAME,
"android");

        driver = new AndroidDriver(new
URL("http://127.0.0.1:4723/wd/hub"),cap);

        driver.get("http://google.com");
        driver.findElement(By.name("q")).sendKeys("Hello Appium !!!");

        driver.quit();

    }

}
```

External dependencies

 # install from HEAD to get important updates

brew install libimobiledevice --HEAD

 # only works for ios 9. for ios 10, see below

brew install ideviceinstaller

There is also a dependency, made necessary by Facebook's WebDriverAgent, for the Carthage dependency manager. If you do not have Carthage on your system, it can also be installed with Homebrew

brew install carthage

ideviceinstaller doesn't work with iOS 10 yet. So we need to install ios-deploy

npm install -g ios-deploy

-----FOR REAL DEVICES-----

For real devices we can use xcpretty to make Xcode output more reasonable. This can be installed by:

sudo gem install xcpretty

Download Android SDK

Download XCODE

export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_131.jdk/Contents/Home

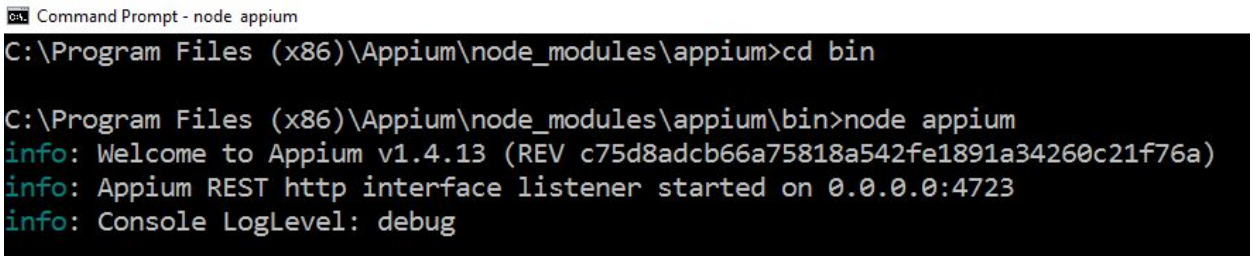
export ANDROID_HOME=/Users/rahularora/Library/Android/sdk

export MAVEN_HOME=/Users/rahularora/Documents/apache-maven-3.3.3/

export PATH=$PATH:$JAVA_HOME/bin:$MAVEN_HOME/bin

export PATH=/Users/rahularora/Documents/apache-maven-3.3.3/bin:$PATH:$MAVEN_HOME/bin:$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools

Navigate to the bin folder in command prompt where Appium is installed C:\Program Files (x86)\Appium\node_modules\appium\bin



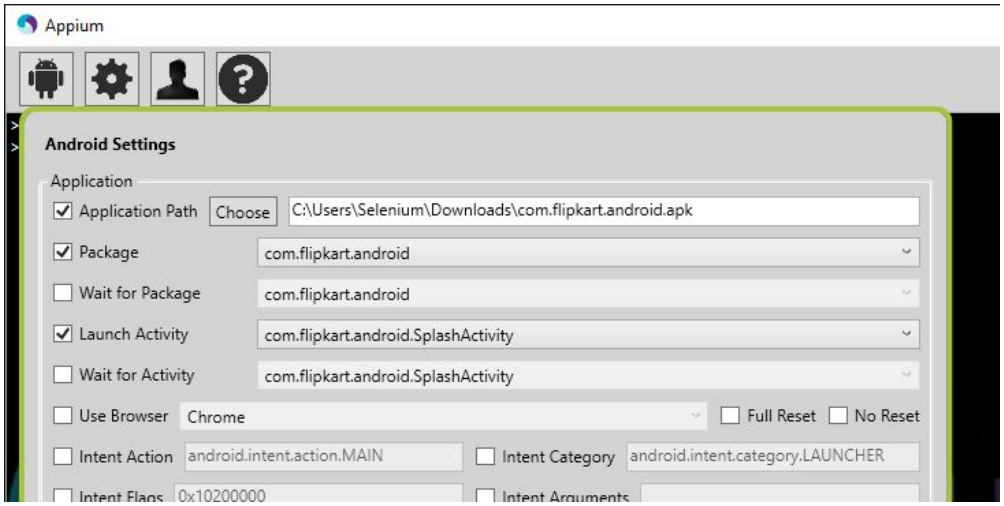
3: Through Java Code:

```
AppiumDriverLocalService service = AppiumDriverLocalService
                    .buildService(new AppiumServiceBuilder()
                    .usingDriverExecutable(new File("c:/Program
Files/nodejs/node.exe"))
                    .withAppiumJS(new File("C:/Program Files
(x86)/Appium/node_modules/appium/bin/appium.js"))
                    .withLogFile(new
File("c:/appiumlogs/logs.txt")));

service.start();

//To stop the services

service.stop();
```

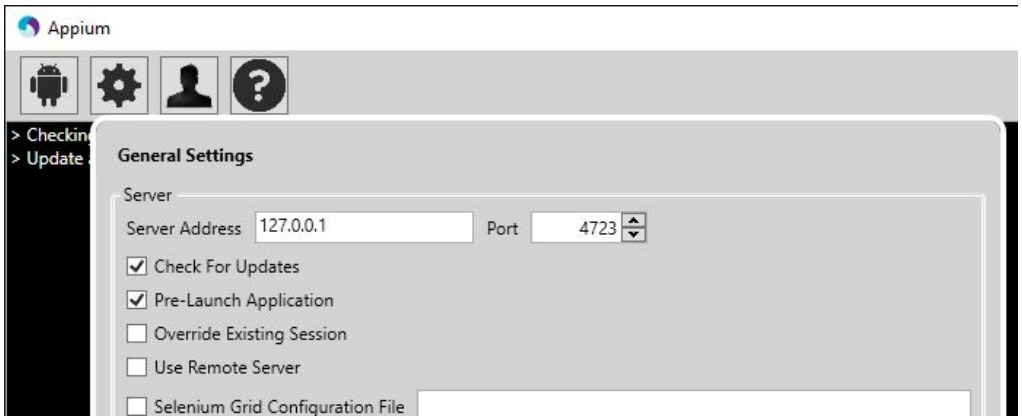
DEPLOYING APPS

1: Through Appium App

a) Get the capabilities from Android Settings



b) Check the Pre-launch under general settings and hit RUN while the device is connected



2: Deploying APK files through code

```
File app = new
File("C:\\Users\\Selenium\\Downloads\\selendroid-test-app-0.17.0
.apk");

          DesiredCapabilities capabilities = new
DesiredCapabilities();

capabilities.setCapability(CapabilityType.BROWSER_NAME, "");
          capabilities.setCapability("device", "Android");
          capabilities.setCapability("deviceName", "Galaxy
Note3");
          capabilities.setCapability("platformVersion", "5.0");
          capabilities.setCapability("platformName","Android");
```

```
          capabilities.setCapability("app",
app.getAbsolutePath());
```

3: Launching already deployed app through Java code

```
DesiredCapabilities capabilities = new DesiredCapabilities();

capabilities.setCapability(CapabilityType.BROWSER_NAME, "");
          capabilities.setCapability("device", "Android");
          capabilities.setCapability("deviceName", "Galaxy
Note3");
          capabilities.setCapability("platformVersion", "5.0");
          capabilities.setCapability("platformName","Android");


//Give the App package and Activity info to launch the already
//installed app

          capabilities.setCapability("appPackage",
                    "com.whatsapp");
capabilities.setCapability("appActivity",
                    "com.whatsapp.Main");
```

chrome://inspect/#devices

Basic Test for Launching WebApp on Chrome Browser

```
public class BasicWebAppTest {
```

```
	public static AndroidDriver<MobileElement> driver;


	public static void main(String[] args) throws
MalformedURLException, InterruptedException {

		AppiumDriverLocalService service =
AppiumDriverLocalService
				.buildService(new AppiumServiceBuilder()
				.usingDriverExecutable(new File("c:/Program
Files/nodejs/node.exe"))
				.withAppiumJS(new File("C:/Program Files
(x86)/Appium/node_modules/appium/bin/appium.js"))
				.withLogFile(new
File("c:/appiumlogs/logs.txt")));

service.start();


		DesiredCapabilities capabilities = new
DesiredCapabilities();
		capabilities.setCapability(CapabilityType.BROWSER_NAME,
"Chrome");
		capabilities.setCapability("device", "Android");
		capabilities.setCapability("deviceName", "Galaxy Note3");
		capabilities.setCapability("platformVersion", "5.0");
		capabilities.setCapability("platformName","Android");

		driver = new AndroidDriver<MobileElement>(new
URL("http://127.0.0.1:4723/wd/hub"), capabilities);
		driver.manage().timeouts().implicitlyWait(20L,
TimeUnit.SECONDS);

		driver.navigate().to("http://way2automation.com");
		System.out.println(driver.getTitle());



		Thread.sleep(3000);



		driver.quit();
service.stop();
	}

}
```

Basic Test for Launching Native / Hybrid App on a Real Device / Emulator when app is already installed

```
DesiredCapabilities capabilities = new DesiredCapabilities();

capabilities.setCapability(CapabilityType.BROWSER_NAME, "");
          capabilities.setCapability("device", "Android");
          capabilities.setCapability("deviceName", "Galaxy
Note3");
          capabilities.setCapability("platformVersion", "5.0");
          capabilities.setCapability("platformName","Android");

          //Give Package Name and Launchable Activity of the APK
file
          capabilities.setCapability("appPackage",
                    "com.whatsapp");
capabilities.setCapability("appActivity",
                    "com.whatsapp.Main");


          driver = new AndroidDriver<MobileElement>(new
URL("http://127.0.0.1:4723/wd/hub"), capabilities);
          driver.manage().timeouts().implicitlyWait(20L,
TimeUnit.SECONDS);
```

Native methods from Android Driver class

Unlocking the device:

```
if(driver.isLocked()){
          driver.unlockDevice();

     }
```

```
driver.unlockDevice();
```

Locking the device:

```
driver.lockDevice();
```

Get Current running activity:

```
System.out.println(driver.currentActivity());
```

Android Key Codes:

http://developer.android.com/reference/android/view/KeyEvent.html

```
driver.pressKeyCode(153);
driver.pressKeyCode(151);
driver.pressKeyCode(145);
```

This will enter the number as 971

Open Notifications:

```
driver.openNotifications();
```

Installing, Removing and Switching Apks

```
//Verify if app exists
```

```
if(!driver.isAppInstalled("io.selendroid.testapp")){

        //Installing the App

driver.installApp("C:\\Users\\Selenium\\Downloads\\sel
endroid-test-app-0.17.0.apk");
        System.out.println("App installed
successfully");


    //Switching between Apks
        driver.startActivity("io.selendroid.testapp",
".HomeScreenActivity");

    }else{

        //Removing the App
        driver.removeApp("io.selendroid.testapp");
        System.out.println("App removed
successfully");
    }
```

More Native commands:

```
driver.pressKeyCode(AndroidKeyCode.BACK);

driver.rotate(ScreenOrientation.LANDSCAPE)

driver.runAppInBackground(10)
```

Native Test – Dialing a Number

```
DesiredCapabilities capabilities = new DesiredCapabilities();
     capabilities.setCapability(CapabilityType.BROWSER_NAME,
"");
     capabilities.setCapability("device", "Android");
     capabilities.setCapability("deviceName", "Galaxy Note3");
     capabilities.setCapability("platformVersion", "5.0");
     capabilities.setCapability("platformName","Android");

     capabilities.setCapability("appPackage",
               "com.android.contacts");
     capabilities.setCapability("appActivity",

"com.android.contacts.activities.PeopleActivity");

     driver = new AndroidDriver<MobileElement>(new
URL("http://127.0.0.1:4723/wd/hub"), capabilities);
     driver.manage().timeouts().implicitlyWait(20L,
TimeUnit.SECONDS);

     Thread.sleep(3000);


driver.findElement(By.xpath("//android.widget.TextView[contains(
@text,'Keypad')]")).click();


driver.findElement(By.id("com.android.contacts:id/nine")).click(
);

driver.findElement(By.id("com.android.contacts:id/seven")).click
();

driver.findElement(By.id("com.android.contacts:id/one")).click()
;

driver.findElement(By.id("com.android.contacts:id/one")).click()
;

driver.findElement(By.id("com.android.contacts:id/one")).click()
;
```

```
driver.findElement(By.id("com.android.contacts:id/one")).click()
;

driver.findElement(By.id("com.android.contacts:id/one")).click()
;

driver.findElement(By.id("com.android.contacts:id/five")).click(
);

driver.findElement(By.id("com.android.contacts:id/five")).click(
);

driver.findElement(By.id("com.android.contacts:id/eight")).click
();


driver.findElement(By.id("com.android.contacts:id/dialButton")).
click();
```

Native Test – Scroll to a contact (UIAutomation)

```
    public static AndroidDriver<MobileElement> driver;

        public static void main(String[] args) throws
IOException, InterruptedException {


        DesiredCapabilities capabilities = new
DesiredCapabilities();

capabilities.setCapability(CapabilityType.BROWSER_NAME, "");
        capabilities.setCapability("device", "Android");
        capabilities.setCapability("deviceName", "Galaxy
Note3");
        capabilities.setCapability("platformVersion", "5.0");
        capabilities.setCapability("platformName","Android");

        capabilities.setCapability("appPackage",
                  "com.android.contacts");
        capabilities.setCapability("appActivity",
```

```
"com.android.contacts.activities.PeopleActivity");
          driver = new AndroidDriver<MobileElement>(new
URL("http://127.0.0.1:4723/wd/hub"), capabilities);
          driver.manage().timeouts().implicitlyWait(20L,
TimeUnit.SECONDS);


System.out.println(driver.findElements(By.id("com.android.contac
ts:id/tab_custom_layout")).size());


driver.findElements(By.id("com.android.contacts:id/tab_custom_la
yout")).get(3).click();
          //First Way
          //driver.scrollTo("Akash").click();


     //Scrolling
          driver.findElementByAndroidUIAutomator("new
UiScrollable(new
UiSelector().scrollable(true).instance(0)).scrollIntoView(new
UiSelector().text(\""+str+"\").instance(0))").click();
```

Native – Android UI Automation

```
File app = new
File(System.getProperty("user.dir")+"\\apk\\selendroid
-test-app-0.17.0.apk");

          DesiredCapabilities capabilities = new
DesiredCapabilities();
```

```
capabilities.setCapability(CapabilityType.BROWSER_NAME
, "");
        capabilities.setCapability("device",
"Android");
        capabilities.setCapability("deviceName",
"Galaxy Note3");
        capabilities.setCapability("platformVersion",
"5.0");

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

        driver = new AndroidDriver<MobileElement>(new
URL("http://127.0.0.1:4723/wd/hub"), capabilities);
        driver.manage().timeouts().implicitlyWait(20L,
TimeUnit.SECONDS);


driver.findElementByAndroidUIAutomator("UiSelector().r
esourceId(\"io.selendroid.testapp:id/my_text_field\")"
).sendKeys("Hello Appium !!!");


    //UIAutomation - checkable

System.out.println(driver.findElementsByAndroidUIAutom
ator("UiSelector().checkable(true)").size());


        //UIAutomation - instance


driver.findElementByAndroidUIAutomator("UiSelector().c
lassName(\"android.widget.CheckBox\").instance(0)").cl
ick();
```

Chrome Browser WebApp Testing

chrome://inspect#devices

Defining Type