APPIUM is a freely distributed open source mobile application UI \_ Testingframework.

Appium allows native, hybrid and web application testing and supports automation test on physical devices as well as on emulator or simulator both.

It offers cross-platform application testing i.e. single API works for both Android and iOS platform test scripts.

It has NO dependency on Mobile device OS. Because, APPIUM has framework or wrapper that translate [Selenium](http://www.guru99.com/selenium-tutorial.html) Webdriver commands into UIAutomation (iOS) or UIAutomator (Android) commands depending on the device type not any OS type.

Appium supports all languages that have Selenium client libraries like- Java, Objective-C, [JavaScript](http://www.guru99.com/interactive-javascript-tutorials.html) with node.js, PHP, Ruby, Python, [C#](http://www.guru99.com/c-tutorial.html) etc.

#### **APPIUM Design Concepts :**

* Appium is an 'HTTP Server' written using [Node.js](http://www.guru99.com/node-js-tutorial.html) platform and drives iOS and Android session using Webdriver JSON wire protocol. Hence, before initializing the Appium Server, [Node.js](http://www.guru99.com/node-js-tutorial.html) must be pre-installed on the system.
* When Appium is downloaded and installed, then a server is setup on our machine that exposes a REST API.
* It receives connection and command request from the client and execute that command on mobile devices (Android / iOS).
* It responds back with HTTP responses. Again, to execute this request, it uses the mobile test automation frameworks to drive the user interface of the apps. Framework like:-
  + Apple Instruments for iOS (Instruments are available only in Xcode 3.0 or later with OS X v10.5 and later)
  + Google UIAutomator for Android API level 16 or higher
  + Selendroid for Android API level 15 or less

**Pre-requisite**

First of all, we need to install a number of software/tools on our machine before we can actually begin automating Android applications:

1. Java Development Kit (JDK)
2. Android SDK
3. Node.js
4. Microsoft Webdriver
5. PDANet+
6. Appium
7. GenyMotion
8. ADT Plugin
9. Java Client Drivers
10. Appium Client Libraries\

The following depicts the in-detail process of installing an Appium environment on your Windows machine for the very first time :-

1. Download and install Java (JDK) and set path of jdk and bin folder

a) Download the “.exe” file from <http://www.oracle.com/technetwork/java/javase/downloads/index.html> (Version: jdk1.8.0\_91 or whichever is the latest you find there.)

b) Install the “.exe” file.

c) Set JDK folder path in your system’s environment variable.

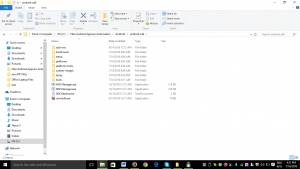
d) Set JDK bin folder path in your system’s environment variable.

2. Download and install Android SDK

a) Download Android SDK from <https://developer.android.com/studio/index.html>

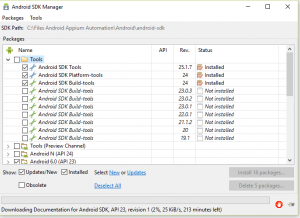
b) Click on the link “android-sdk\_r24.4.1-windows.zip” (or whichever is the latest you find there) and then click on the download button.

c) Once a zip file gets downloaded, unzip the folder. You will get the following list of folders inside the Android SDK.



d) Now click on the “SDK Manager.exe” file.

e) This opens the Android SDK Manager window. Select the first 3 packages under “Tools”, select the package under Tools (Preview Channel) and finally the Android platform based on the platform of the device on which you will be performing your tests (Android 6, in my case).



Only after all the selected packages are installed you would be able to write and execute your test scripts.(Please note: This could take a few hours depending on your internet speed

3. Set SDK path in Windows

After all the packages are successfully installed, you need to set the SDK path.

a) Click on Environment Variables -> Create a new user variable ANDROID\_HOME -> and set the Android SDK path as the value for it. (e.g.: C:\Files Android Appium Automation\Android\android-sdk)

b) Set 2 paths in the system variable Path:

i) Path of “platform tools” folder in the SDK (e.g.: C:\Files Android Appium Automation\Android\android-sdk\platform-tools)

ii) Path of “tools” folder in SDK (e.g.: C:\Files Android Appium Automation\Android\android-sdk\tools)

Now, to check whether or not Android is configured properly in your system, run command “android” in the command prompt. This will open the Android SDK Manager dialogue box, which verifies successful configuration of Android in your system.

4. Node.js

Follow these steps:-

a) Go to the link <https://nodejs.org/en/download/>

b) Click on the Windows Installer tab.

c) Download starts.

d) Install it.

5. Microsoft Webdriver

Follow these steps:-

a) Go to the link <https://www.microsoft.com/en-us/download/details.aspx?id=48212>

b) Click on the Download link on this page.

c) Install it.

6. PDANet+

To download and install PDANet+ on your machine, follow these steps:-

a) Go to the link [pdanet.co/](http://pdanet.co/)

b) Download and install it.

7. Appium

This is a test automation tool for mobile applications.

a) Go to the link [appium.io](http://appium.io/)

b) Click on “AppiumForWindows.zip” link from [here](https://bitbucket.org/appium/appium.app/downloads/).

c) Download starts.

d) Unzip the donwloaded zipped folder.

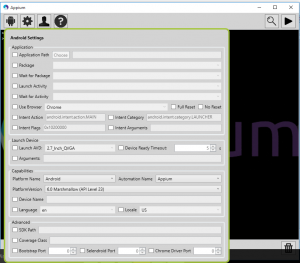
e) Install the exe file “appium-installer”.

Configure Appium

Android Settings:

a) Click on Android Settings button in the top left corner of Appium window.

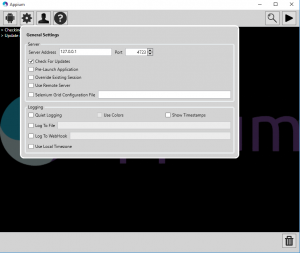
b) Select Platform Name as Android, Automation Name as Appium and Platform version as your device’s OS version.



General Settings:

a) Click on the General Settings button.

b) Just note down the Server Address and the Port Number. You’ll be needing it during automation script creation.



SETUP FOR AUTOMATING ANDROID APPLICATIONS USING ANDROID VIRTUAL DEVICE(OR EMULATOR): You can even run your test scripts on a virtual device or an emulator, instead of running it on a real device. But before doing that, there are some further installations you need to make (apart from the above mentioned ones).

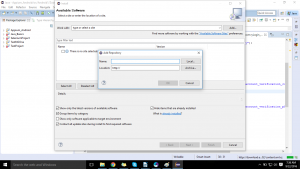
9. ADT Plugin: It is used to access Android SDK within Eclipse. ADT Plugin tool is basically used to launch Android Emulators in Eclipse using the test script that we have created. By installing this plugin you can basically launch AVD Manager from Eclipse.

Steps to install ADT Plugin in Eclipse:

a) Open Eclipse IDE.

b) Go to Help -> Install New Software. This will open Install Software dialogue.

c) Click on the add button here. This will open a new dialogue.



d) Set URL “https://dl-ssl.google.com/android/eclipse/” in location text box and click on OK button.

e) It will open Developer tools option along with a checkbox.

f) Select it and click on the Next button.

g) Click on the next button from here, accept the Terms & Conditions and click on the Finish button.

h) This will start installing the Plugin.

i) After successful installation, restart Eclipse.

Set SDK Location: You need to set the SDK folder path after installing ADT Plugin

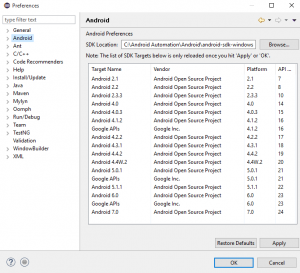
a) Open Eclipse.

b) Go to Window -> Preferences

c) Select “Android” from the list on the left

d) Set SDK folder path in the “SDK Location” box.

e) Click on “OK” after this.

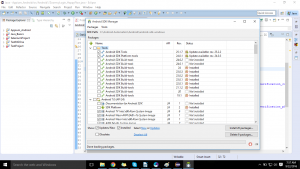


To verify if Android SDK is integrated properly or not

a) Open Eclipse IDE.

b) Go to Window -> Android SDK Manager.

c) This will open the Android SDK Manager dialogue.



This verifies that Android SDK has been properly integrated with Eclipse IDE.

Now, before you can begin creating and running your own test scripts using Appium, there are some configurations that you need to make for your Java Project in Eclipse.

10. Eclipse Configuration with Selenium WebDriver: This is needed for the interaction between your test scripts and the Selenium WebDriver. For the same, you will be needing language-specific client drivers. Since we will be working on Java, we will be needing the Java client drivers.

a) Go to this link: <http://docs.seleniumhq.org/download/>

b) Click on the Download link here for the latest JAR available (selenium-2.53.0 in my case)

c) Extract the downloaded zipped folder

d) Open Eclipse IDE

e) Create a new Java Project -> Create a new package under this project -> Create a new class under this package

f) Right click on your Project name -> Select Build Path -> Select Configure Build Path

g) Click on “Add External JARs” button -> go to the path where you had saved the Selenium WebDriver zip folder (e.g. in my case: C:\Android Automation\selenium-java-2.53.0\selenium-2.53.0)

h) Select both .jar files from here.

i) Now select all the .jar files inside the libs folder here.

Don’t close the Properties Dialogue, since there are a few more JARs which need to be added to your project.

11. Eclipse Configuration with Appium:

a) Go to this link: <http://appium.io/downloads.html>

b) Click on the Java link under the Appium Client Libraries section.

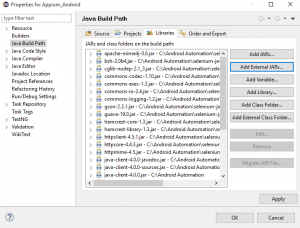
c) Click on the JAR link here.

d) Download starts.

e) Follow the same steps as in 10. to import Appium client libraries into your Eclipse project.

f) Click on OK after importing all the JAR files.

So this is how it’ll look:



Conclusion:

You can put all the prerequisites needed for starting with Appium in a single folder (Android Automation, in my case) which will make it easily accessible. This is what you will end up with:

