Setting Up Your Test Automation Environment

By Ken Asanion

For any questions or clarifications regarding the instructions I provided, contact me at kenasanion@gmail.com

Disclaimer: I'm currently an OSX user, so there might be some lapses regarding the steps that involves Windows such as environmental variables. I've set up a guide for it.

Introduction

For our training, we would utilize Eclipse as our IDE. Basically, you could use other variants of IDEs like Android Studio. However, for all of us to be on the same page on our future test automation exercises, we would use Eclipse IDE. Eclipse is a powerful IDE for Java and is both available on Windows and OSX. For our exercises in automation, we would not create complicated programs, but we will use this IDE to manage our libraries and file structure with ease. Android Studio can also be used but for simplicity purposes, Eclipse would do.

First things first

We would be needing the following files in order to start our own test automation project. Like what was discussed earlier, we would use Appium. The following files are needed:

- ✓ Latest **Java SDK** and **JRE**
- ✓ Latest **Android SDK and Emulator**
- ✓ An IDE like **Eclipse** or Android Studio
- ~~✓ NodeJS (No need for newer version of Appium)~~
- ✓ .NET Framework latest version (Not applicable for OSX)
- ✓ **Appium Server** (Link: http://appium.io)
- ✓ **Appium jar library files** (Link: http://appium.io/downloads)
- ✓ **Selenium jar files** (Link: http://www.seleniumhq.org/download/)
- ✓ **Commons-Lang3 jar file** (https://mvnrepository.com/artifact/org.apache.commons/commons-lang3/3.7)

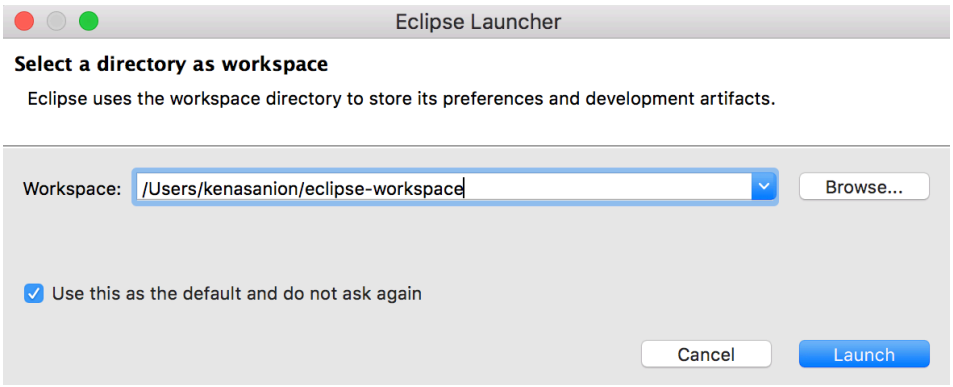
To ensure you have your Java in your system, open a terminal app or command prompt and type in `java –version`. This would show you the Java version as well as your JRE. For Android studio, it already comes with an Android SDK and emulator so you don't need to worry if you selected the standard installation. If you are opting to use Eclipse, you need to manually install the Android SDK archive in their site. We would also be needing the SDK Command line (CLI) tools.

```
Kens-MacBook-Pro:~ kenasanion$ java -version
java version "1.8.0_102"
Java(TM) SE Runtime Environment (build 1.8.0_102-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.102-b14, mixed mode)
Kens-MacBook-Pro:~ kenasanion$
```

Preparing our environment for Appium in Windows / OSX

For **OSX** users, you may also need to install the latest version of ruby using Brew (Link: https://www.ruby-lang.org/en/documentation/installation/ - Look for Homebrew (OSX) section. In addition, configuring environmental variables should be done in the bash_profile.

1. Download and Install the latest **Java SDK bundled (SDKs are bundled with JRE)**
   - SDK: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
2. Set **JAVA_HOME** to installation jdk folder (Environmental Variables)
   - Watch the video at 1:00 minute mark to know how. For OSX, these are already setup.
3. Download and Install **Android SDK**
   - Android Studio and CLI Tools: https://developer.android.com/studio/index.html
4. Set **ANDROID_HOME** to SDK installation directory (Environmental Variables)
5. Download and install latest **.NET Framework** version (https://www.microsoft.com/net/download/dotnet-framework-runtime)
6. Download latest version Eclipse.
   - As of writing, the latest stable version is **Oxygen**. https://www.eclipse.org/downloads
7. Install Eclipse. Once finished, set the default directory for your workspace where all of your projects will be saved.



OSX ONLY: Instructions via Terminal

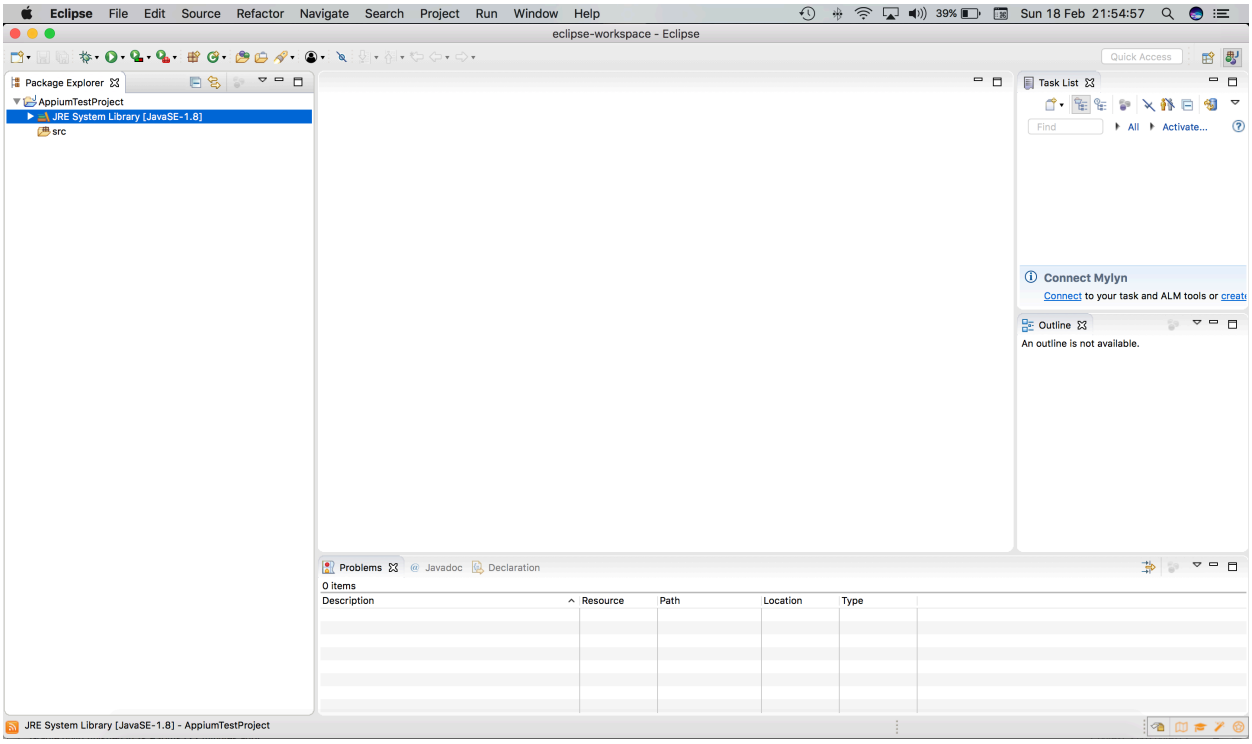
```
nano ~/.bash_profile

export ANDROID_HOME=/YOUR_PATH_TO/android-sdk
export PATH=$PATH:$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools

source ~/.bash_profile
```

---

use `echo $ANDROID_HOME` to verify

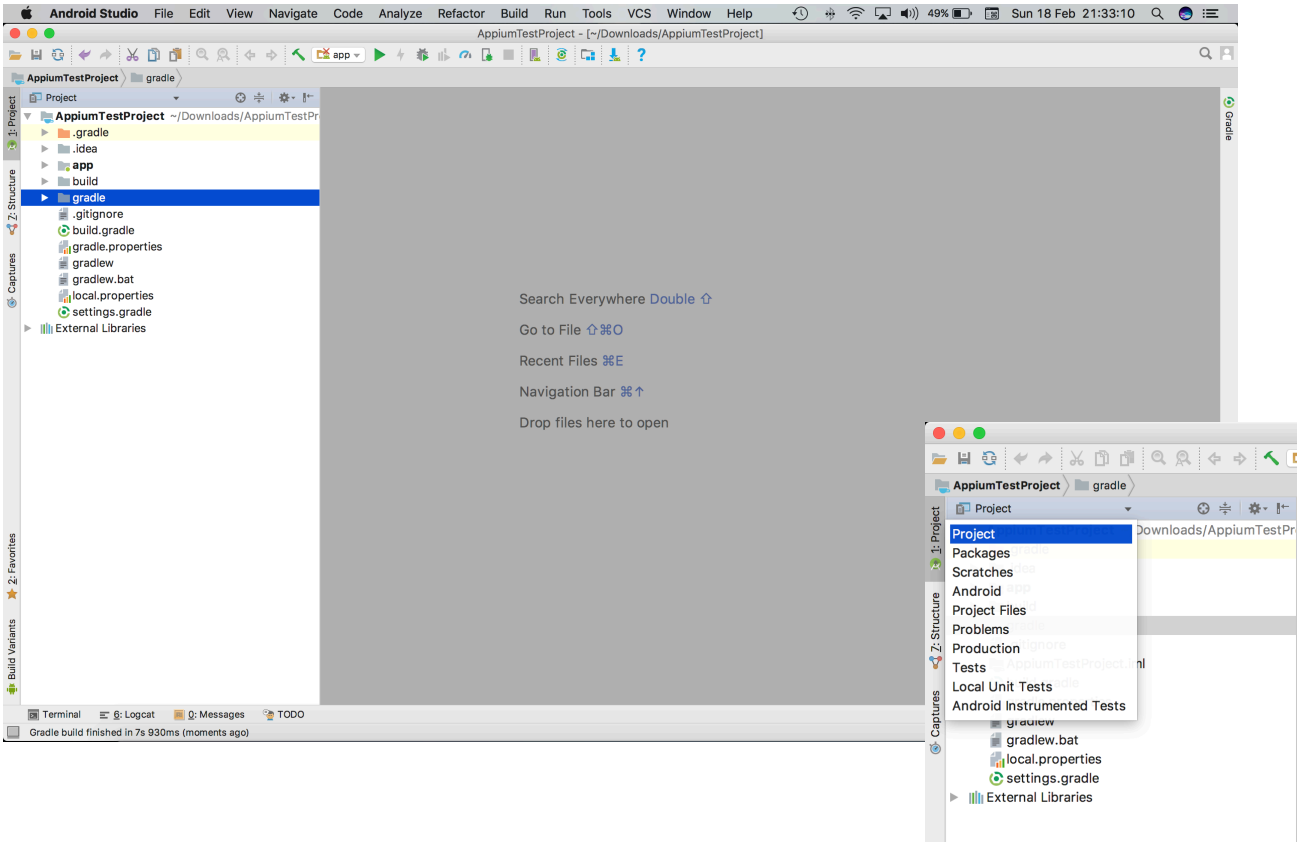
Result
Eclipse:

Most of you who followed the steps above would see a screen like this. It's pretty simple compared to Android Studio and is enough to cater our scripting needs.



Optional step in case you opted to set everything up in Android Studio instead of Eclipse

Since we will not be creating an Android application, we would want our project explorer to use the Project view instead of the Android option. You can change your view by clicking the dropdown located on the top explorer bar and choose `Project`.



If you want to know how to setup libraries in Android studio which I will not cover, you may do so by checking this link https://www.youtube.com/watch?v=n8QqYAXKWEU.

Downloading and Installing Appium

1. Download latest client for Selenium for Java http://www.seleniumhq.org/download/ (Appium is based on top of Selenium. We would be needing its client libraries)
2. Download the Appium client library http://appium.io/downloads.html. We need this for developing our test scripts. Save it on your working directory for later use.
3. Download latest version of **Appium Server** (https://github.com/appium/appium-desktop)
    a. GitHub download link: https://github.com/appium/appium-desktop/releases/
        i. For OSX, download Appium-1.3.2.dmg
        ii. For Windows, download Appium.Setup.1.3.2.exe
    b. **Optional** - For curiosity purposes, source codes are also downloadable since this is an open source project
4. Download the latest Commons Lang 3 - https://mvnrepository.com/artifact/org.apache.commons/commons-lang3/3.7. This is for us to not encounter errors when using Appium specific drivers.

Note: If you want to see a visual of how these steps are executed, go to this link: https://www.youtube.com/watch?v=OJhKxfTBiWU up until 25:40. The one in the video is the old version of Appium.

Setup Android Debug Bridge (adb) in our CLI

This part is optional. However, it would be wise and faster if we would add the `adb` command as part of our CLI. We would be needing this in the future when dealing with our emulated or physical devices.

- Windows : https://www.howtogeek.com/118594/how-to-edit-your-system-path-for-easy-command-line-access/
- OSX : https://stackoverflow.com/a/32314718
    - If you were able to set up your ANDROID_HOME paths earlier, you may skip this step.

Phew! That was a lot of stuff, but once we setup everything right, we can now start developing our Appium test scripts.

If you still have trouble installation, don't fret to ask or contact me via email.

After you are done setting up everything, you can now open and understand the next chapter which is **Getting Started with Appium**.