Andrew's Appium & XCode Set up on Mac

****
Do not install anything with the sudo command
****

1. Install XCode if not already installed on your computer. To check:

   $ xcode-select -p
   (If you see something like "/Applications/Xcode.app/Contents/Developer", then XCode was already installed.)

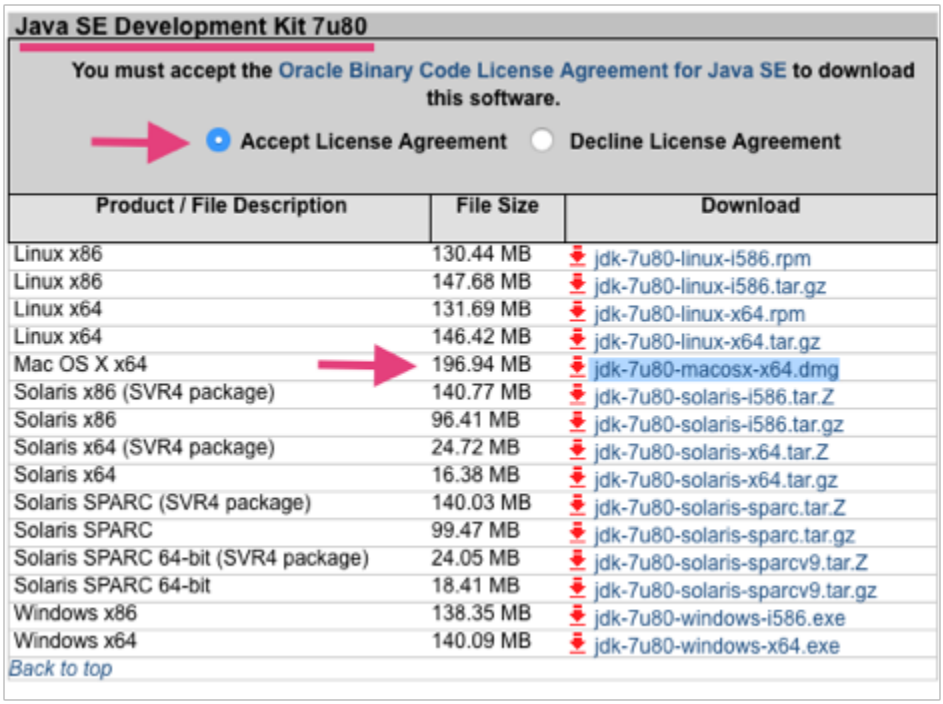
   If Xcode was not installed, install it from Apple's App Store --> Apple App Store

2. Install Command Line Tools if not installed:

   $ gcc --help

   If Command Line Tools are installed you will see the help file for gcc displayed.
   If you don't have Command Line Tools, a pop-up will open prompting you to install it

3. Set up Java Development Kit:
   a. Download jdk-7u80-macosx-x64.dmg from this link (http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html#jdk-7u80-oth-JPR) (You might have to set up an account to download it)



Java SE Development Kit 7u80

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux x86	130.44 MB	jdk-7u80-linux-i586.rpm
Linux x86	147.68 MB	jdk-7u80-linux-i586.tar.gz
Linux x64	131.69 MB	jdk-7u80-linux-x64.rpm
Linux x64	146.42 MB	jdk-7u80-linux-x64.tar.gz
Mac OS X x64	196.94 MB	jdk-7u80-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.77 MB	jdk-7u80-solaris-i586.tar.Z
Solaris x86	96.41 MB	jdk-7u80-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.72 MB	jdk-7u80-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u80-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140.03 MB	jdk-7u80-solaris-sparc.tar.Z
Solaris SPARC	99.47 MB	jdk-7u80-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24.05 MB	jdk-7u80-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.41 MB	jdk-7u80-solaris-sparcv9.tar.gz
Windows x86	138.35 MB	jdk-7u80-windows-i586.exe
Windows x64	140.09 MB	jdk-7u80-windows-x64.exe

Back to top

   b. Install the jkd with the default settings
   c. Now, at your termina console:

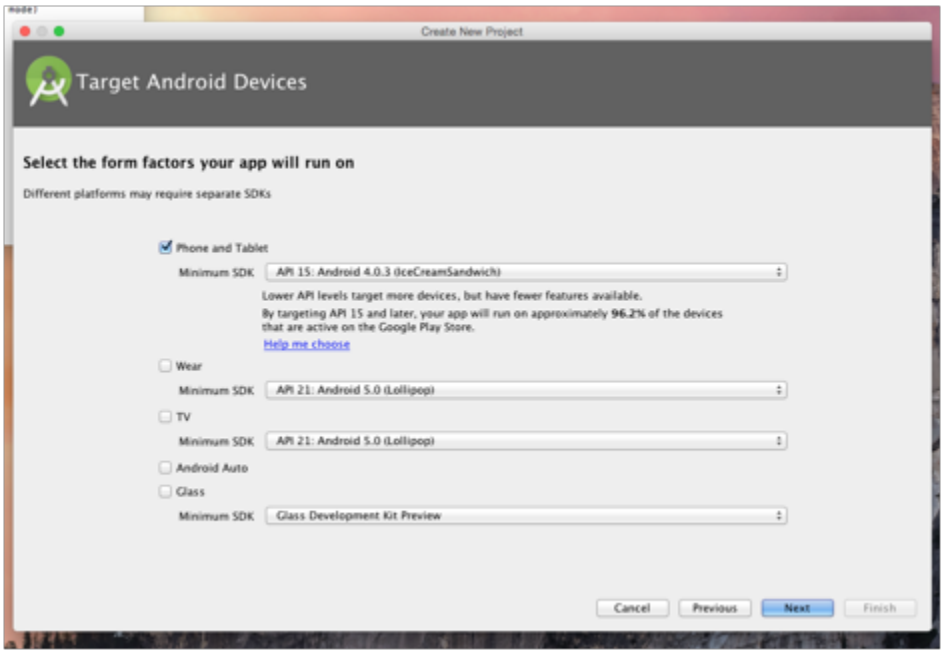
      $ cd
      Copy the following line into your .bash_profile and save:
      export JAVA_HOME=`/usr/libexec/java_home -v 1.7`

4. Close and restart the terminal console. Then check what java version you have:

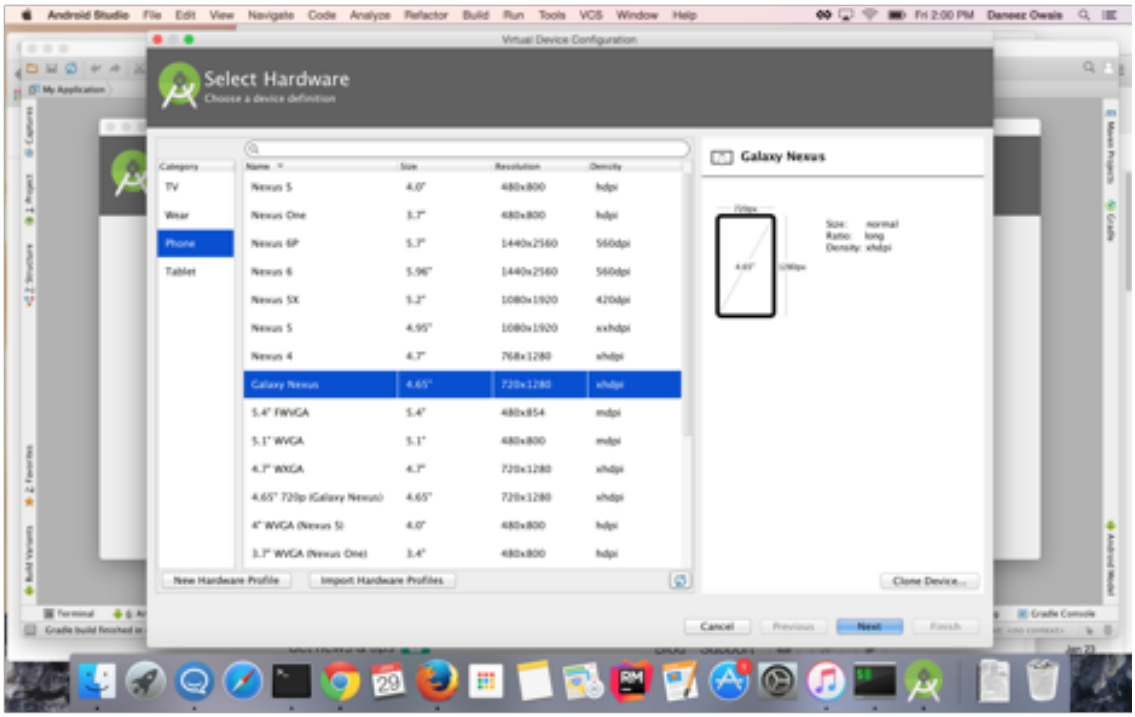
   $ java -version
   (Confirm that you get version 1.7.0_80 or 1.8.0_66)

5. Download Android Studio from this link. (developer.android.com/sdk/index.html)
6. Installing Android Studio:
   a. Launch the downloaded .dmg file
   b. Drag and drop it into your Application folder and eject from devices in finder.
   c. Open Android Studio

d. Start a new project
e. Follow the set up wizard, keeping to the default settings. For project name, just use anything.



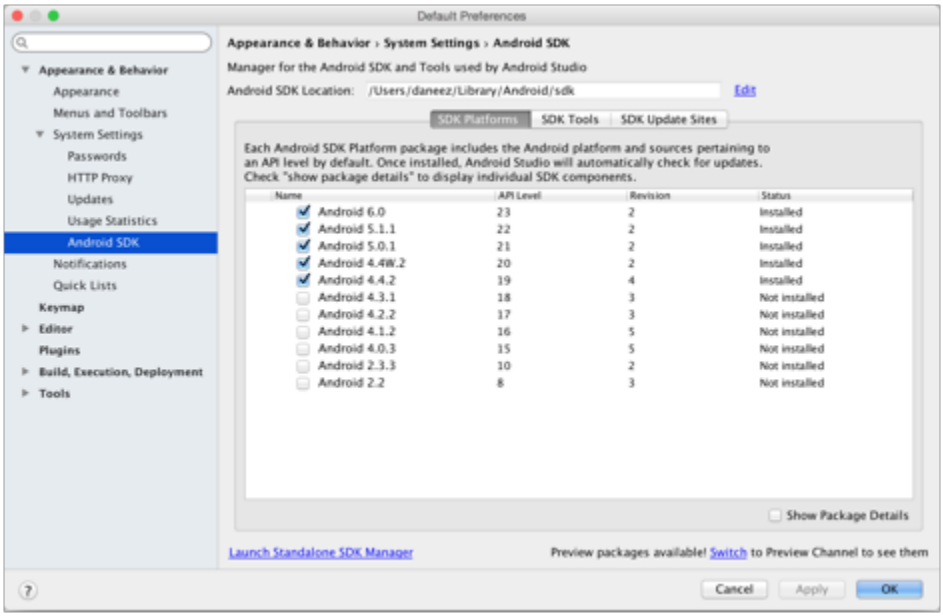
f. Select Blank Activity
g. Set up an Android Virtual Device:
    i. Tools > Android > AVD Manager > Click on create virtual device
    ii. Select Galaxy Nexus as the device



    iii. For system image, click the download link for KitKat, API Level 19, x86
    iv. After it is downloaded, click Next
    v. Name your AVD "Galaxy_Nexus_API_19" and finish
    vi. Click on the play icon to start your device to make sure it is usable.

Setting up the SDK Manager:

1. Tools > Android > SDK Manager. You will see which platform you have installed
2. Check mark all boxes starting with Android 4.4.2 and up and click OK button. This will install all platforms starting with API level 19 and up on your computer. By the end you will see the below screen:



Set up ANDROID_HOME:

1. Open up your terminal console

   $ cd

2. Open up .bash_profile with your favorite editor and paste in these lines:

   export ANDROID_HOME=/Users/<daneez>/Library/Android/sdk
   export PATH=$PATH:$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools
   (replace <daneez> with your username, of course)

3. Restart terminal and type:

   > echo $ANDROID_HOME
   (That should display the path to Android Home - /users/<user>/Library/Android/sdk)

   > adb
   (adb should now work)

4. Installing Homebrew:

   Open your terminal console and type:

   $ brew

   If Homebrew was not installed:

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

5. Install npm and node:

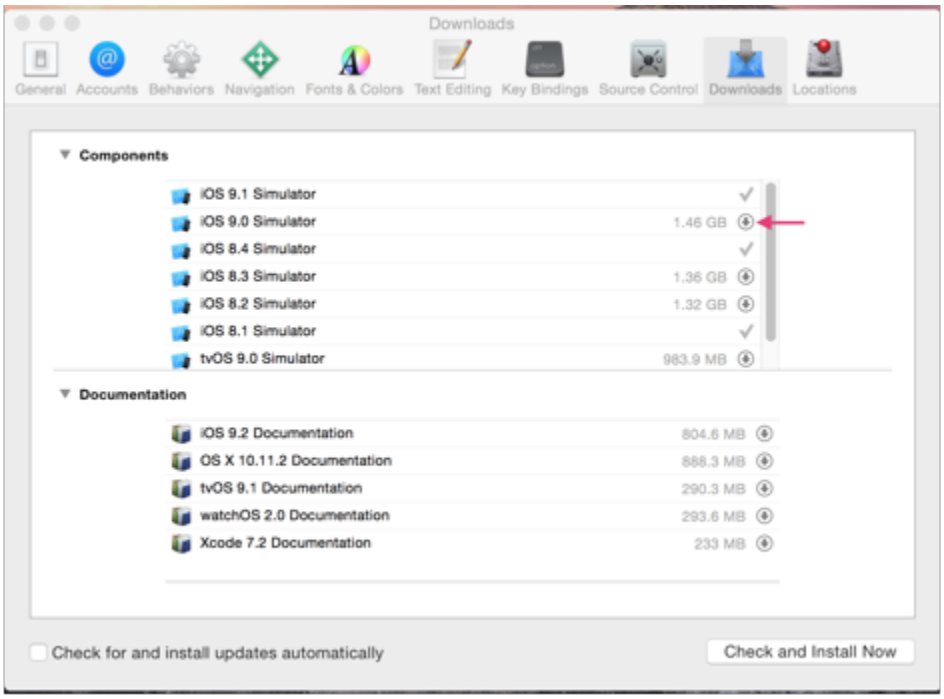
   $ brew install node

   You can test your installation using these commands at the terminal:

   $ node -v
   $ npm -v

6. Downloading simulators in Xcode through the app:

7. Launch Xcode
    a. Install additional required components
    b. Xcode > Preferences in the menu on top left corner. Then click on the Downloads tab.
    c. Click on the little arrows to the right of the iOS 8.4, 9.1, 9.0, or whichever simulator to download the simulator (you might have to provide your password to authorize the download).



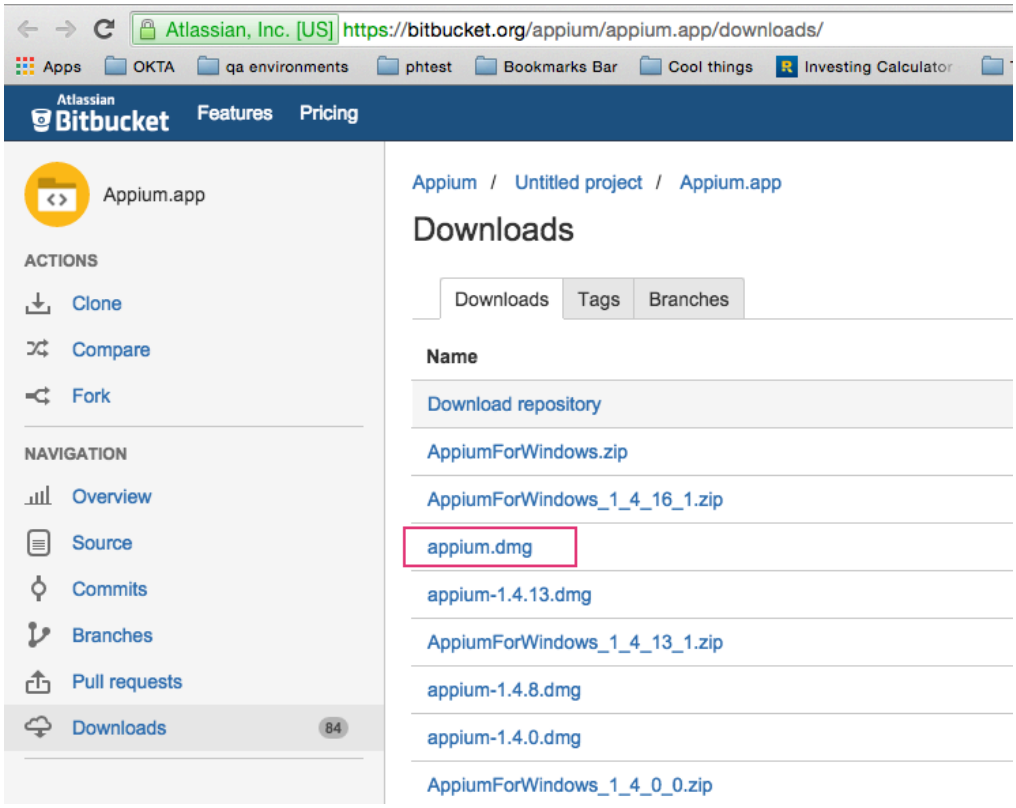
Installing Appium (to use on command line):

$ npm install -g appium

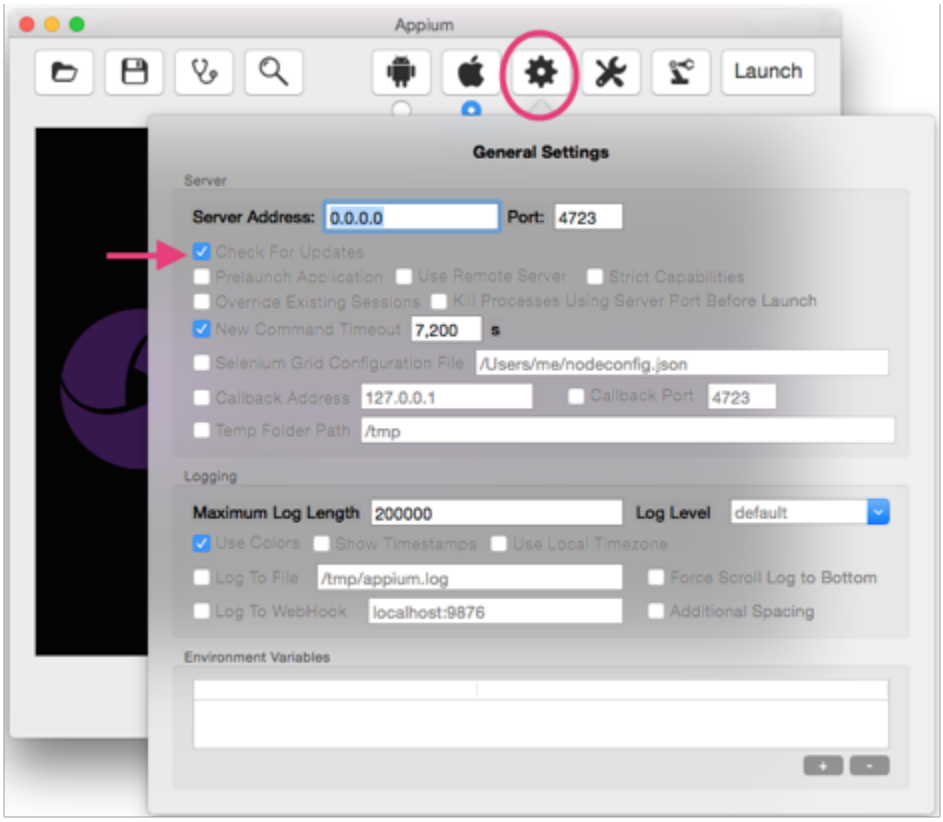
(Check your installation using the command below)
$ appium --version

Installing Appium GUI:

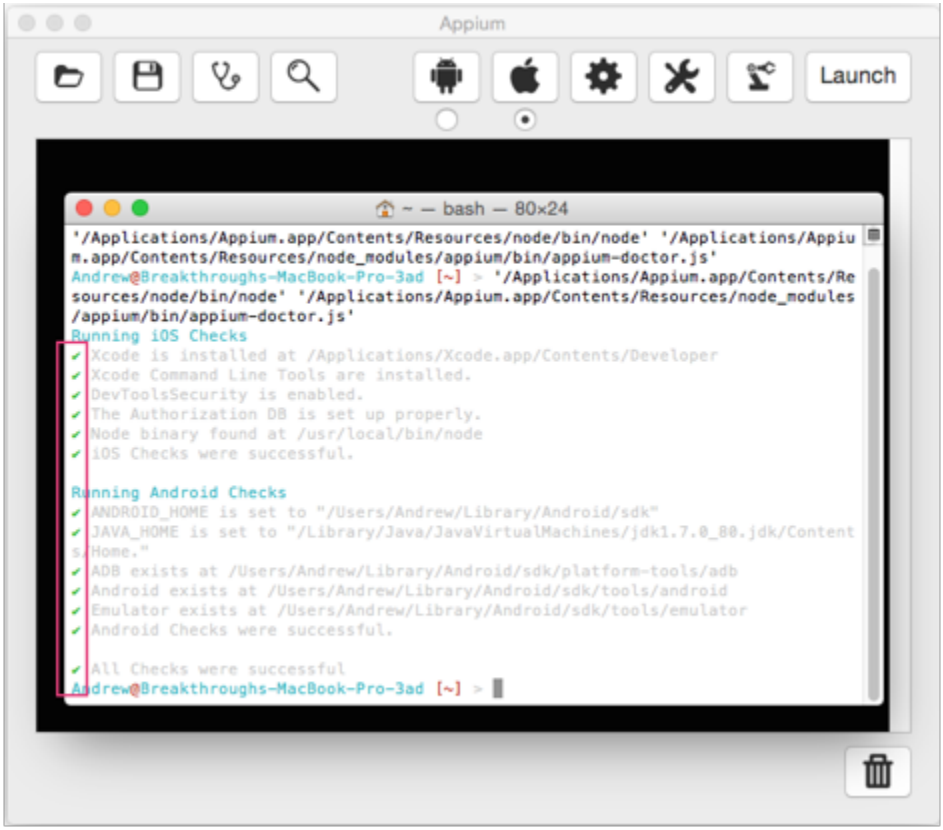
1. Download a copy of appium-1.4.13 or a later version (.dmg) from Appium Bit Bucket



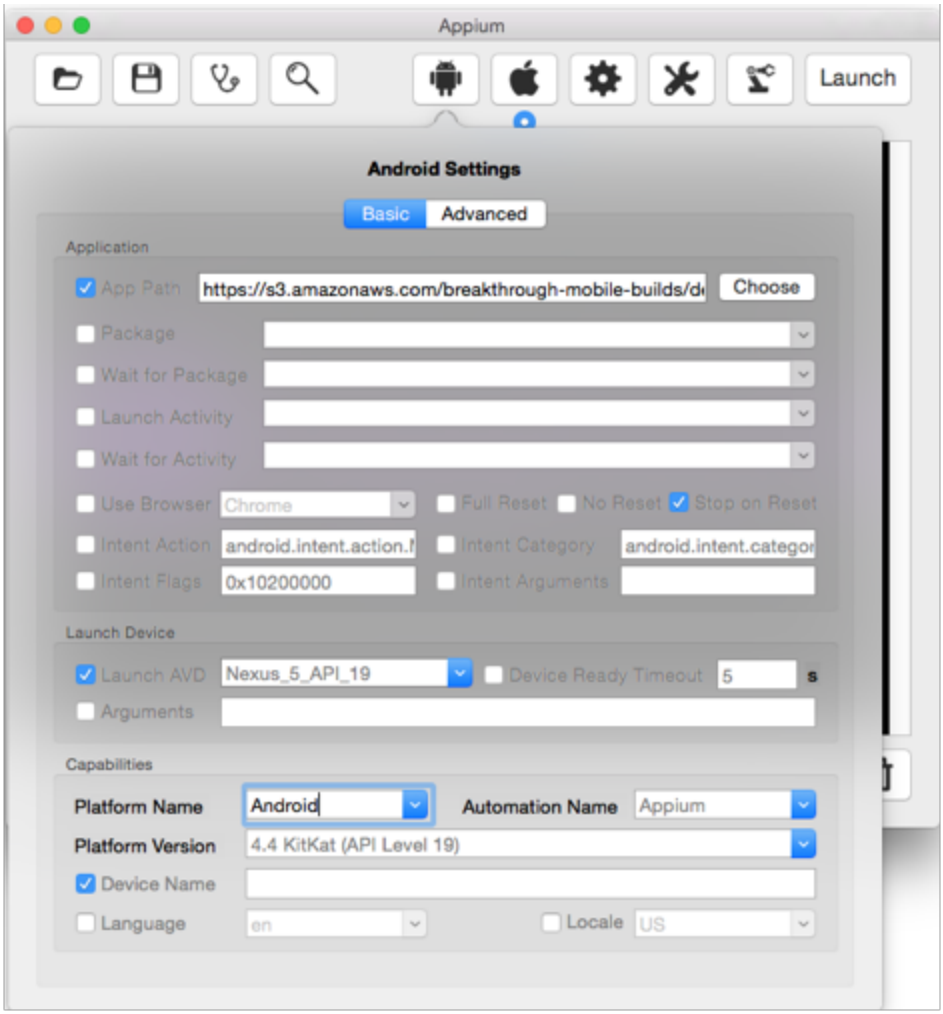
2. Install it by searching for it in firefox or safari (download does not work in Chrome) and add it to the Applications folder
3. Open the app
4. Click on General Settings icon and uncheck the Check For Updates checkbox:



5. Click on the stethoscope icon to run Appium doctor. Make sure all the checks are okay.

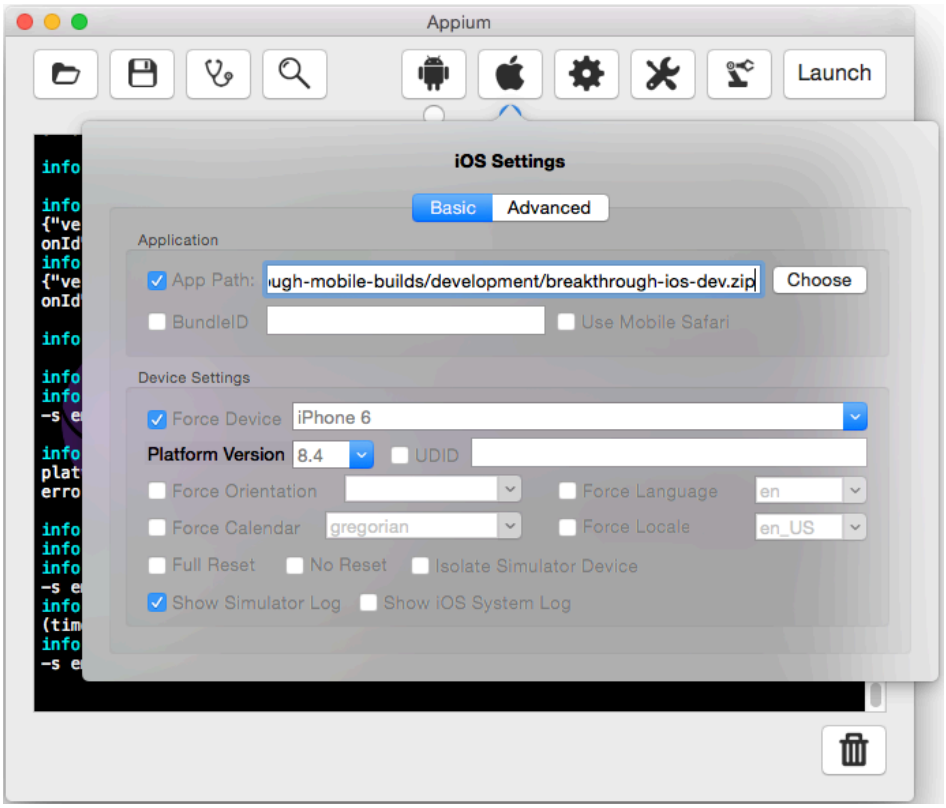


6. Setting up for Android testing:
    a. Click on the Android icon to open up the Android settings
    b. Do the followings:
        i. Set App Path to: https://s3.amazonaws.com/breakthrough-mobile-builds/development/breakthrough-production.apk
        ii. Checkmark App Path.
        iii. Checkmark 'Stop on Reset'
        iv. Checkmark 'Launch AVD' and select 'Galaxy_Nexus_API_19'
        v. Checkmark 'Device Name'
        vi. Platform Name: Android
        vii. Automation Name: Appium
        viii. Platform Version: 4.4 KitKat (API Level 19)

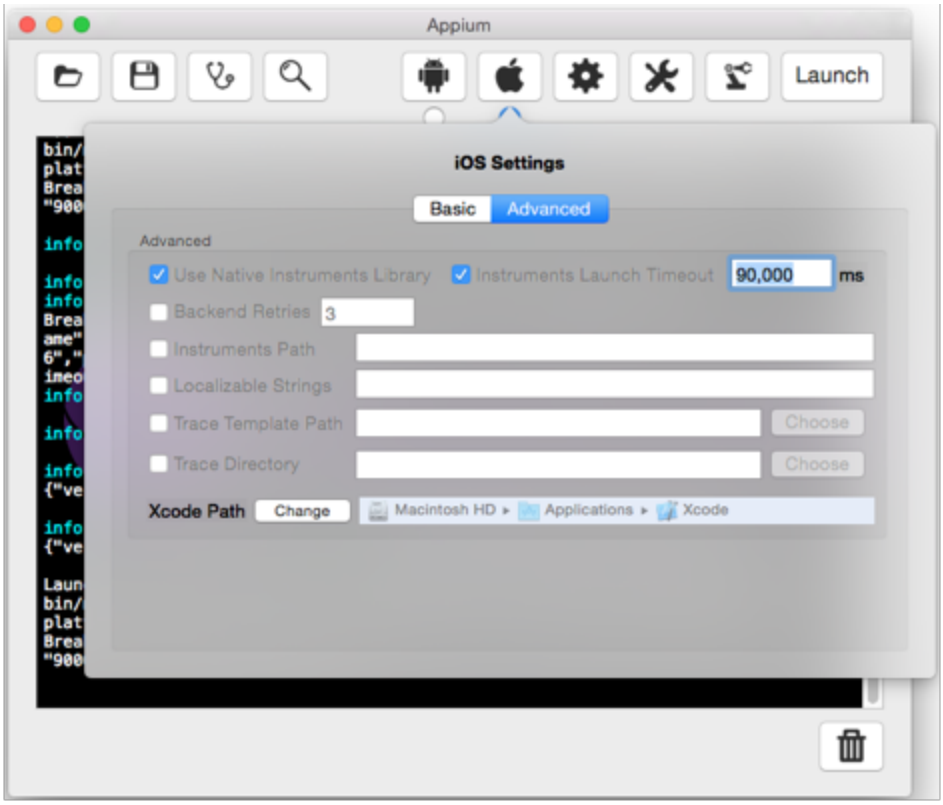


ix. Click on radio button underneath the Android button. Now click the 'Launch' button and wait a few seconds.
x. Click on the Magnifier Glass icon. Wait for a few seconds and your Android Virtual Device (AVD) will launch, download the apk and automatically launch the app.
xi. Once the AVD and app are successfully launched, close the AVD and click on 'Stop' button.

7. Setting up for iOS testing:
    a. Click on the Apple icon and do the following:
        i. Checkmark 'App Path' and enter into the field 'https://s3.amazonaws.com/breakthrough-mobile-builds/development/breakthrough-ios-dev.zip'
        ii. Checkmark 'Force Device' and select iPhone 6
        iii. Select 8.4 as the Platform Version
        iv. Checkmark 'Show Simulator Log'



b. Click on the 'Advanced' tab and do the following:
    i. Checkmark 'Use Native Instrument Library' and 'Instruments Launch Timeout'



8. Launching the inspector to test the settings:
    a. Click on the radio button underneath the Apple icon to select it, then click the 'Launch' button

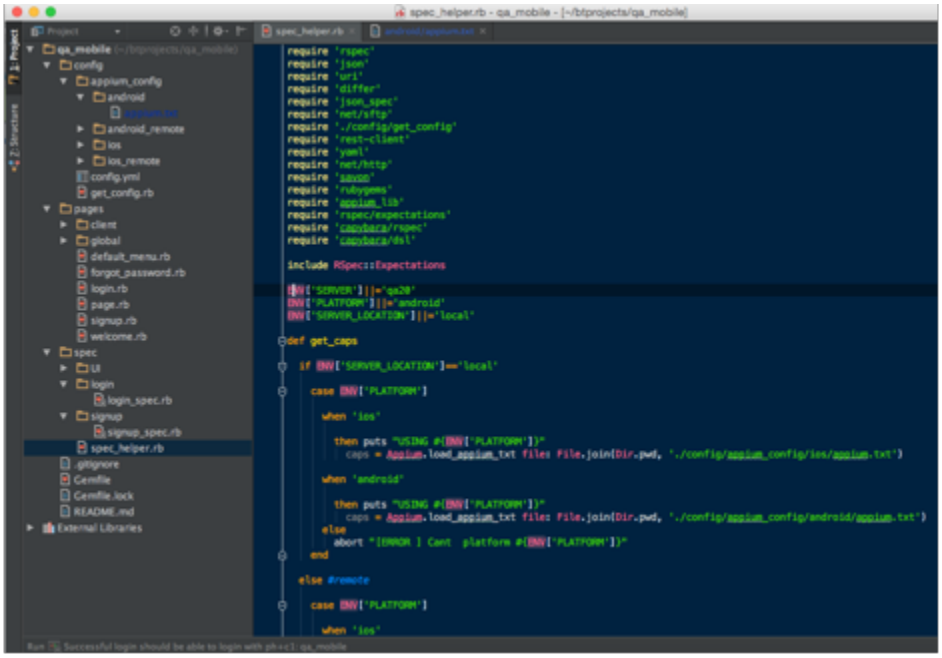
b. Wait a few seconds and click on the Magnifier Glass icon. Wait a little bit for your iPhone simulator to launch and install the app.
c. Once your iPhone simulator and app are launched, you can close the simulator and click 'Stop'.
d. Click on the radio button underneath the Android icon to select it, then click the 'Launch' button
e. Wait a few seconds and click on the Magnifier Glass icon. Wait a little bit for your Android Virtual Device (AVD) to launch and install the app.
f. Once your AVD and app are launched, you can close the AVD and click 'Stop' on the Appium GUI.

Congrats!!! You have successfully set up Appium for inspecting the app on Android and iOS.

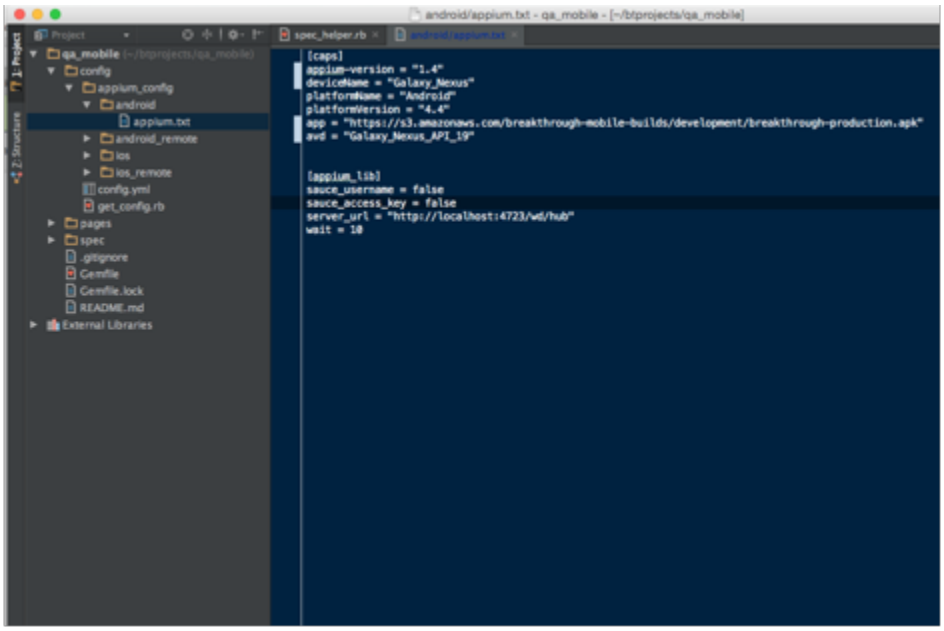
9. Test running mobile automation in RubyMine:
   a. Clone the mobile automation code for BT

      $ git clone git@github.com:BreakthroughBehavioralInc/qa_mobile.git

   b. Open the project in RubyMine and click on the link to install any missing gems.
   c. Open qa_mobile > spec > spec_helper.rb and make sure the platform is 'android' and the server_location is 'local'



   d. Open qa_mobile > config > appium_config > android > appium.txt and change to the below:



e. Now, open qa_mobile > spec > login > login_spec.rb
f. In the terminal console, launch the appium server by typing:

   $ appium (then press 'Enter')

g. Right click on the spec 'should be able to login with ph+c1' and run it (the first Run option)