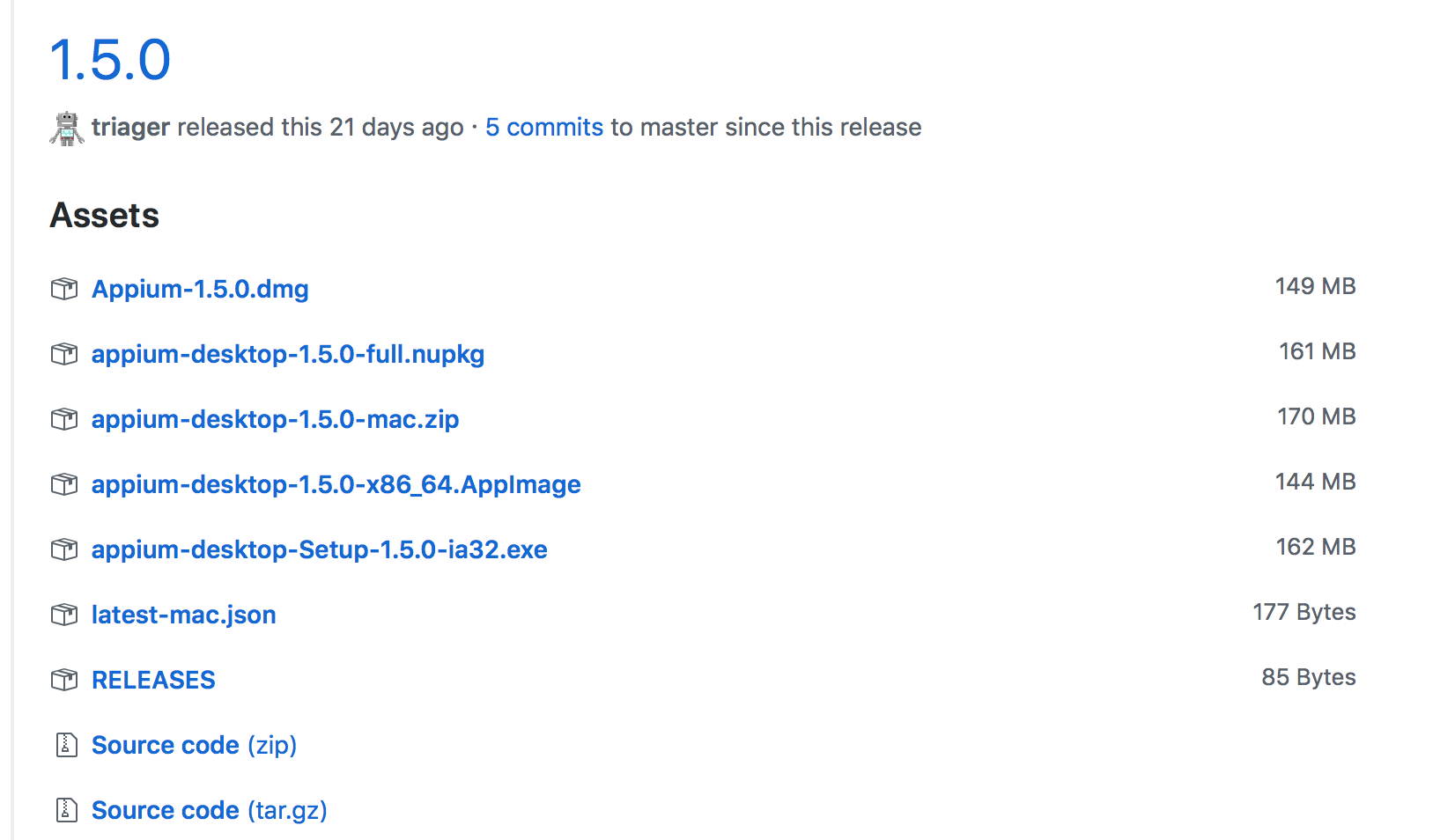
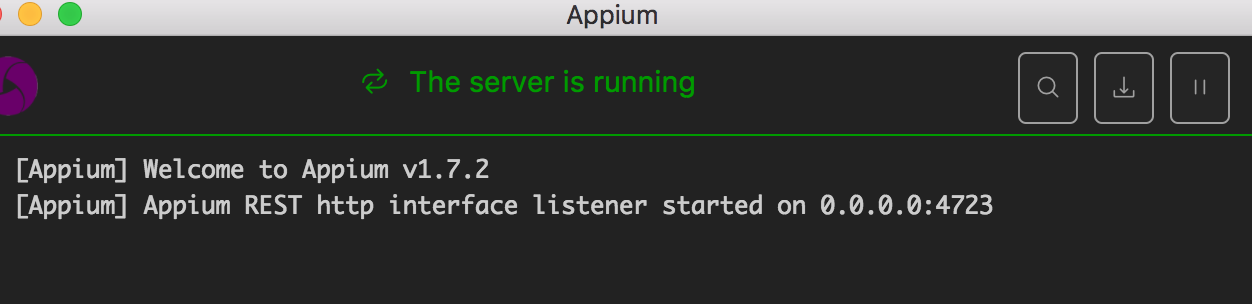
**Mobile (Android/iOS) Testing Automation using Appium**

**Software required and pre-requisites:**

* Appium – download it from <http://appium.io> and click on the download Appium and choose the setup file according to your need. **(Eg. for mac you have to download Appium-1.5.0.dmg)**  
  
* Open the terminal in your mac and run the following commands. For running these command your system must have brew, for installing the brew run the below command:  
  **ruby -e "$(curl –fsSL** [**https://raw.githubusercontent.com/Homebrew/install/master/install)**](https://raw.githubusercontent.com/Homebrew/install/master/install))**"**
* **Run the below commands:**
* brew install carthage
* brew install libimobiledevice --HEAD
* brew install ideviceinstaller
* npm install -g ios-deploy
* gem install xcpretty
* brew update

**Mobile Android automation using Appium:**

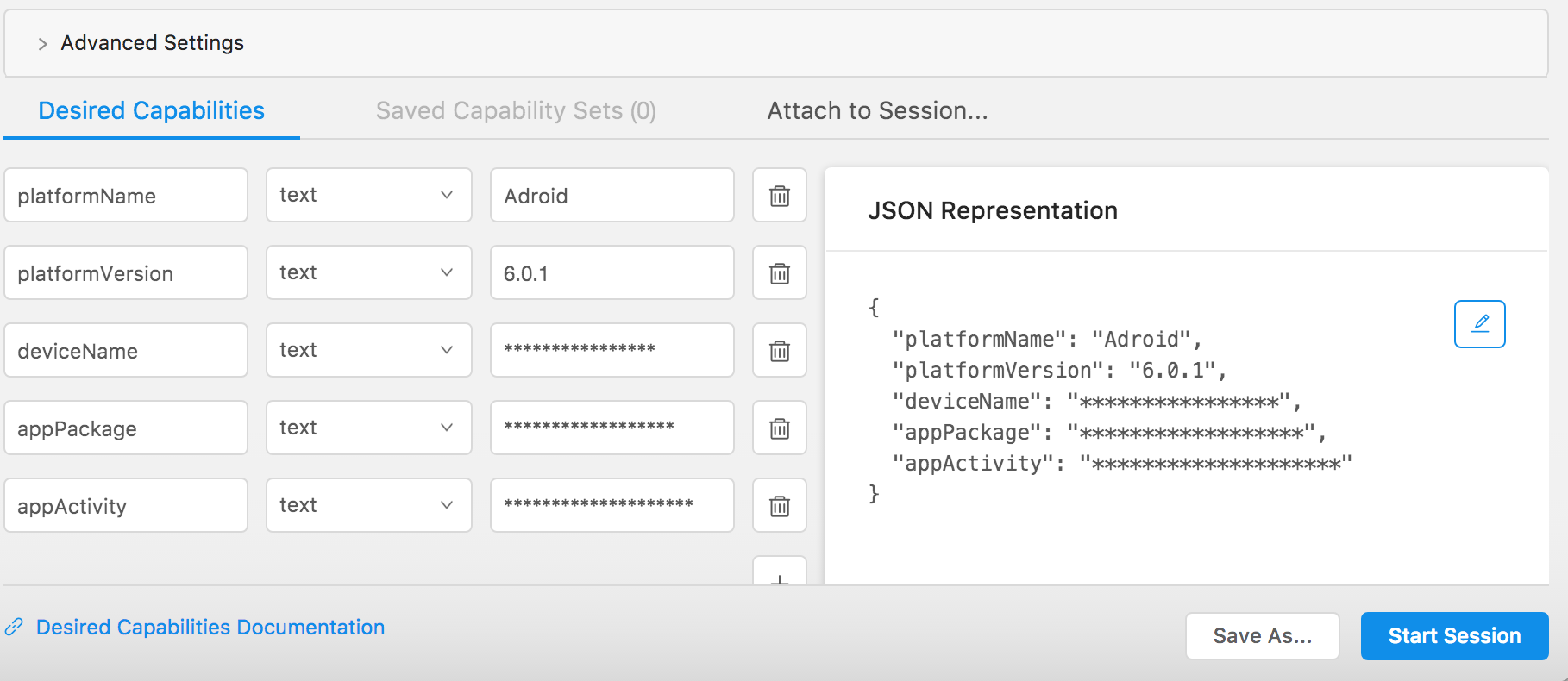
* Open the Appium from **finder 🡪 Application 🡪 Appium.  
  **
* Start the server.
* Once the server is running click on the search button  
  ****
* Defined the desired capabilities:  
  For Android –   
  **{  
   "platformName": "Android",   
   "platformVersion": "6.0.1",**

**"deviceName": "Your device name",**

**"appPackage": "app package name",**

**"appActivity": "app package activity"**

**}**

* These desired capabilities depend on the app which you have installed in your phone
* For getting the device name open the terminal give the command **adb devices.**
* Once all the capabilities are set click on the start session.  
    
  **Due to privacy reason I have hide the deviceName, appPackage and appActivity. Here you will mention yours.**
* **Key things to note:**  
  Your Android version should be higher or equal to 6.0.0.  
  Your app must be installed in the android phone.  
  Device must be connected from system when you are starting the session.

**Mobile iOS Automation using Appium:  
  
Key things to note in iOS Automation using Appium:**

* This will only work for applications that are signed with a **DEVELOPMENT** cert.
* This will **NOT** work for applications that are signed with a **DISTRIBUTION** cert because Apple's Instruments will not allow you to interact with those applications which is live. (Even if you knew the **bundleId**)
* Device should have provisioning profile. Device UDID should be mentioned in the provisioning profile which will be used to sign the app.

# ([How to prepare iOS app for Appium testing for external test team?](http://discuss.appium.io/t/how-to-prepare-ios-app-for-appium-testing-for-external-test-team/802)To build the app with your certificate and then to share that certificate with the rest of the QA team to use when instrumenting the app.)

* You must install the Xcode in you machine

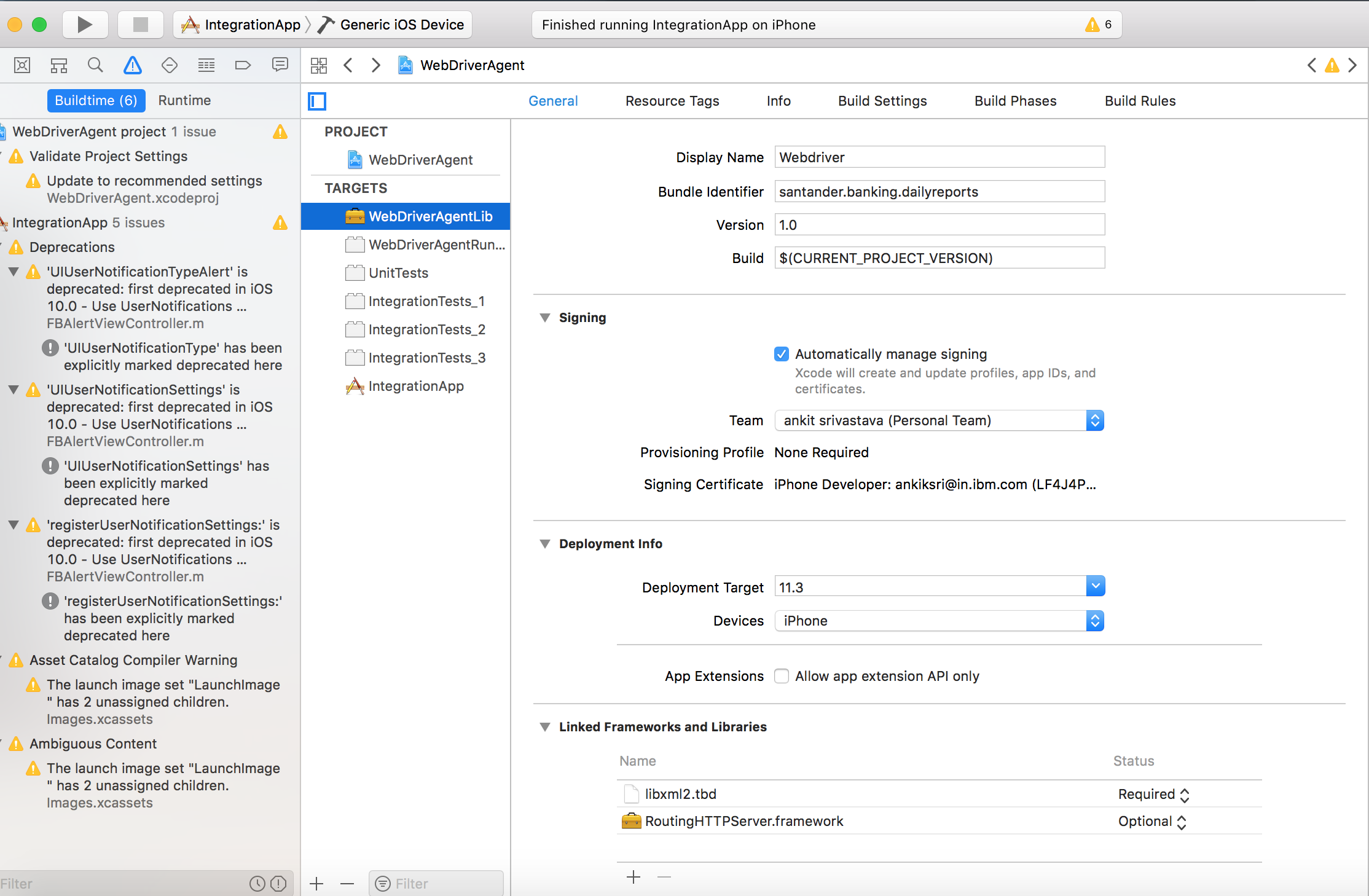
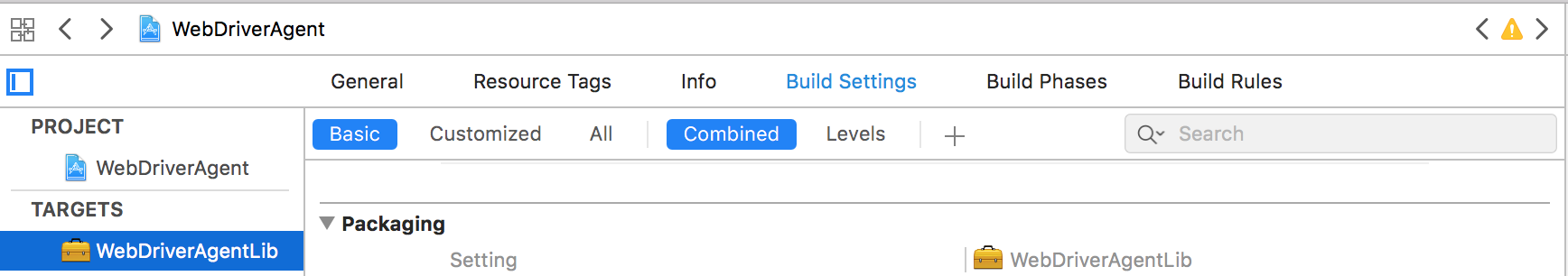
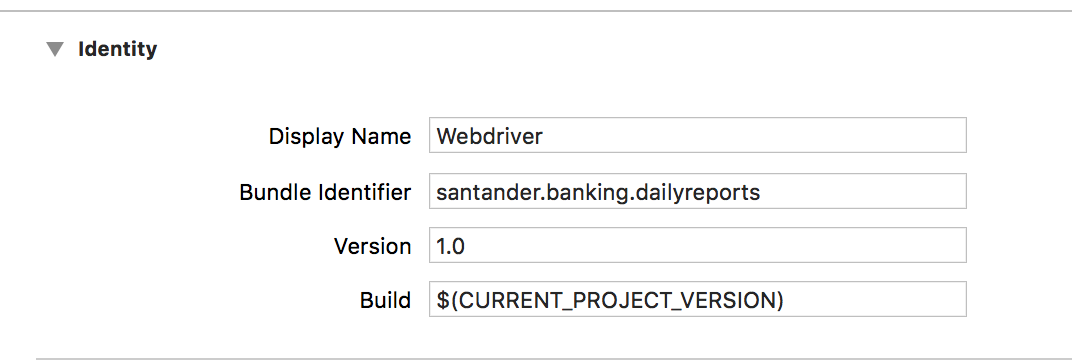
**Commands you have to run:**brew install carthage  
brew install libimobiledevice –HEAD  
brew install ideviceinstaller  
brew install ideviceinstaller

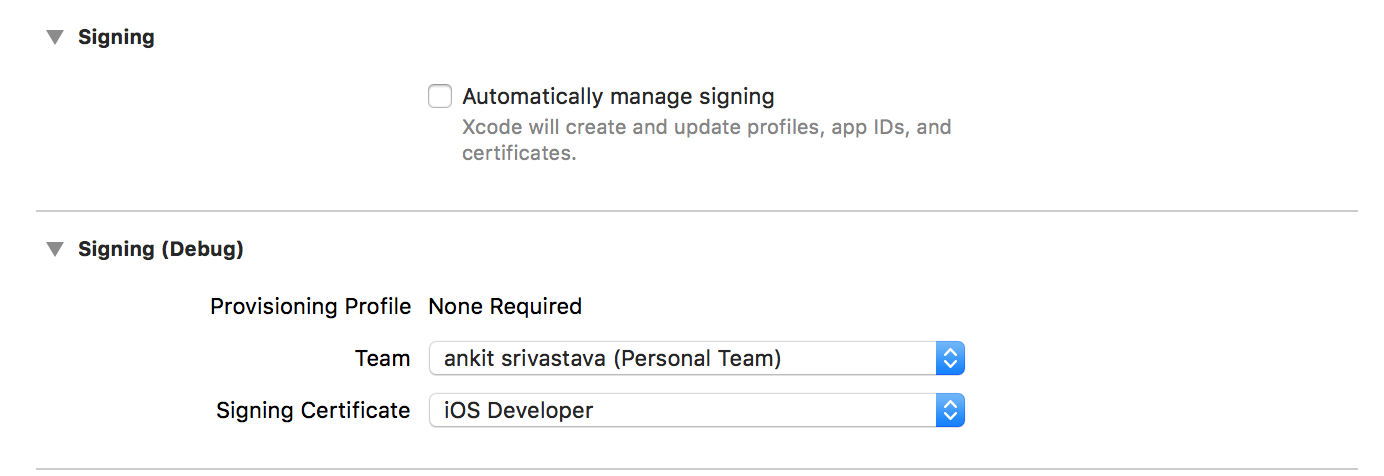
npm install -g ios-deploy  
npm install –g deviceconsole

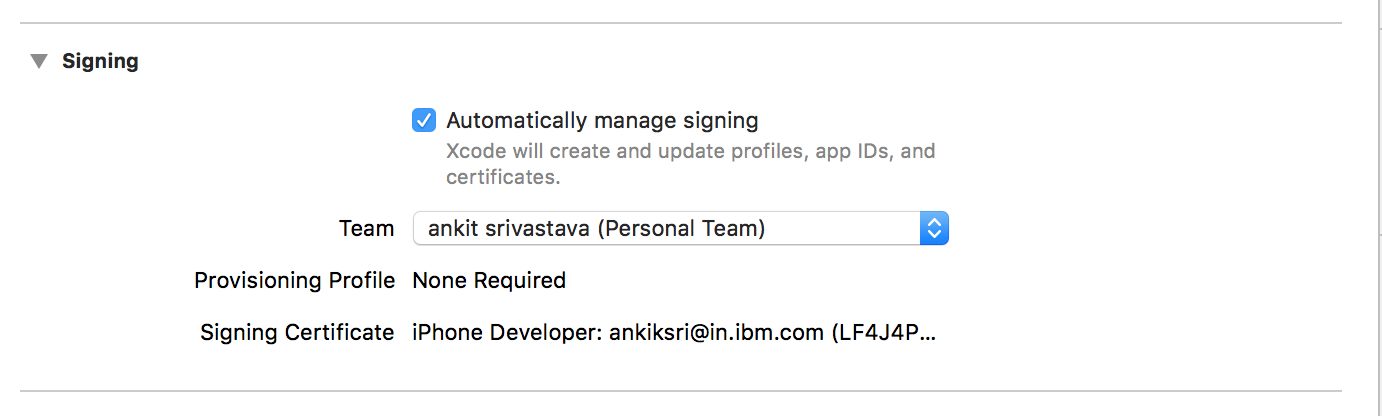
gem install xcpretty

brew update

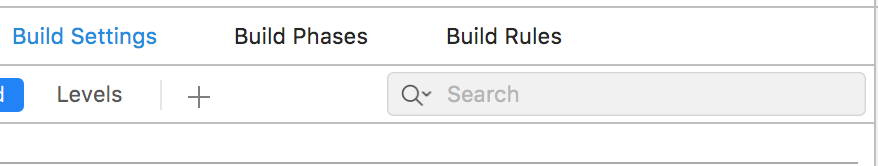
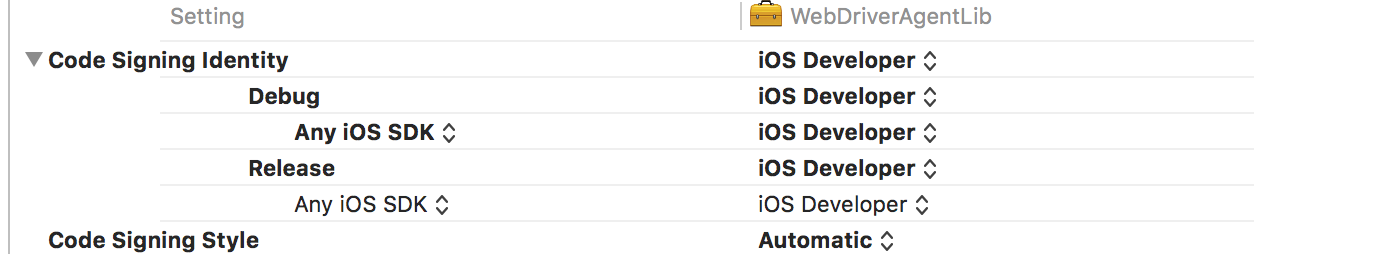
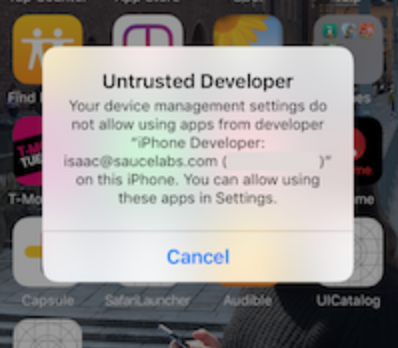
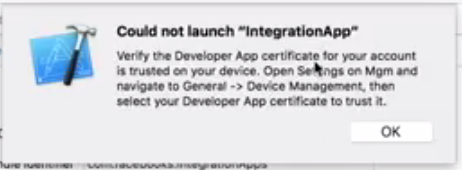
**Real Device configuration:**

* Go in the application Right click on the Appium click on the **show Package content**.
* Then follow this path:  
  **Contents🡪Resources🡪app🡪node\_modules🡪appium🡪node\_modules🡪appium-xcuitest-driver🡪WebDriverAgent**
* Give the command in your terminal   
  **cd /Applications/Appium.app/Contents/Resources/app/node\_modules/appium/node\_modules/appium-xcuitest-driver/WebDriverAgent**
* Once you are inside WebDriverAgent directory give these command
* **mkdir –p Resources/WebDriverAgent.bundle**
* **./Scripts/bootstrap.sh –d**
* Now follow this path:  
  **/Applications/Appium.app/Contents/Resources/app/node\_modules/appium/node\_modules/appium-xcuitest-driver/WebDriverAgent**
* Open WebDriverAgent.xcodeproj  
  
* There you will get **WebDriverAgentLib, WebDriverAgentRunner, Unit test, IntegrationApp**.
* One by one define the **developer certificate, bundleId of the app and provisioning profiles** in **General, info and Build setting**.  
  
* For Eg: I am showing one for “**WebDriverAgentLib**”  
  **General**
* Define the bundle id of the app:  
  
* Choose Signing Option **Manually/Automatic**. If you have your developer certificate and provisioning profile ready and installed choose manual.
* Your **developer certificate and device must be patched in the provisioning profile** if you are using manual.

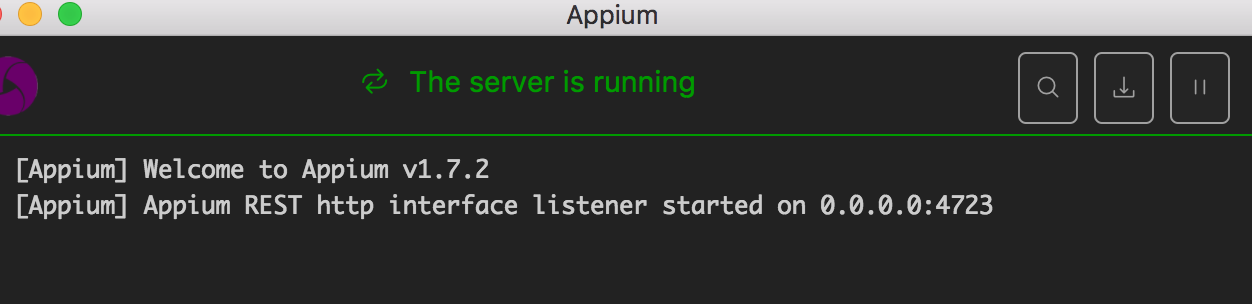


* if you don’t have any **provisioning profile and developer certificate** choose automatic signing it will automatically create the developer certificate and provisioning profile.  
  

**Build Setting:**

* Go to the search option under Build Settings   
  
* search for **code signing identity**Define the developer certificate. if you are using automatic code signing then no need to define it will take automatically.
* Under Build Settings only search for **provisioning profile**.  
    
  Define the provisioning profile, If you are doing automatic code signing then choose automatic otherwise select the **related provisioning profile**.
* Same thing you have to do for all (**WebDriverAgentRunner, Unit test, IntegrationApp)**
* Once the provisioning profile, bundle id and developer certificate are set the open the terminal and give the following command:  
  **cd /Applications/Appium.app/Contents/Resources/app/node\_modules/appium/node\_modules/appium-xcuitest-driver/WebDriverAgent  
    
  xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination ‘id=UDID of real device’ test**
* Don’t worry if the test is failing follow the below thing.
* Go to the following path in your Appium show package content.  
  **Contents🡪Resources🡪app🡪node\_modules🡪appium🡪node\_modules🡪appium-xcuitest-driver🡪WebDriverAgent🡪Configurations🡪ProjectSettings.xcconfig**
* Edit the .xcconfig file and add the following capabilities in that file:  
  **DEVELOPMENT\_TEAM = <’Team ID’>  
  CODE\_SIGN\_IDENTITY = iPhone Developer**
* Team Id you will get from your developer certificate. Open your **keychain access under my certificate you can see the developer certificate**.
* Double click on that and look for   
  **Organizational Unit = ‘\*\*\*\*\*\*\*\*\*\*’  
  Due to privacy reason I can’t share the screen shot.**
* Again run the test command it will be successful.
* If you got this kind of error:  
    
     
   **OR**  
  
* You can go to **Settings** 🡪 **General** 🡪 **Device Management** on the device to trust the developer and allow the WebDriverAgentRunner app to be run.

**Appium Settings**

* Launch the Appium app start the server  
  
* Once the server is running click on the search button  
  ****
* Set the capabilities:  
  {

"platformName": "iOS",

"platformVersion": "11.3",

"deviceName": "Your device name",

"automationName": "XCUITest",

"bundleId": "your app id",

"udid": "your device UDID",

"autoWebview": "true",

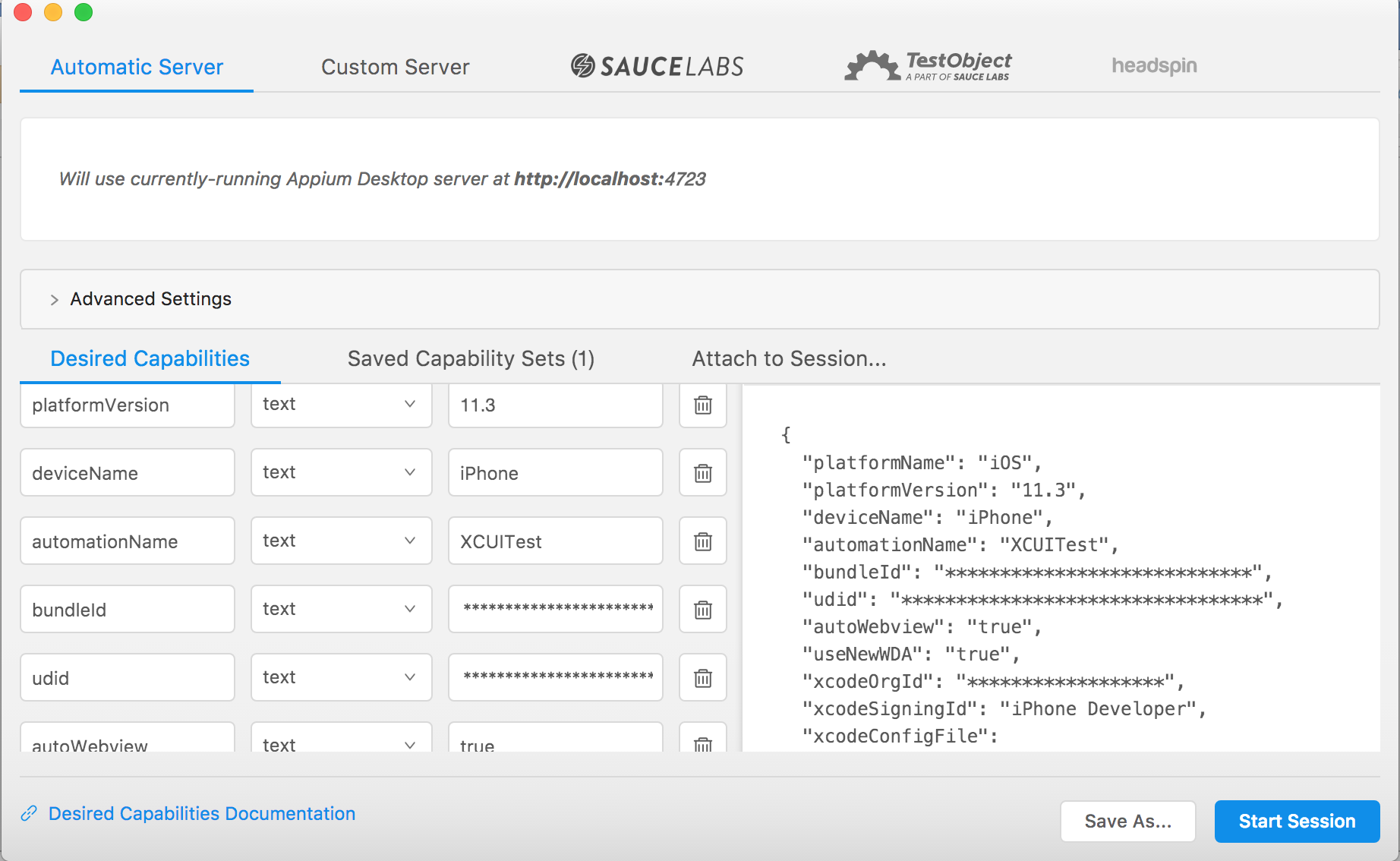
"useNewWDA": "true",

"xcodeOrgId": "Team id",

"xcodeSigningId": "iPhone Developer",

"xcodeConfigFile": "/Applications/Appium.app/Contents/Resources/app/node\_modules/appium/node\_modules/appium-xcuitest-driver/WebDriverAgent/Configurations/ProjectSettings.xcconfig"

}

* For Eg I have defined like this   
    
  **Due to privacy reason I have hide the bundle id, UDID and App id. Here you will mention yours.**
* Start the session you The app will automatically get invoked the phone.