

Mobile Automation - Software Installation

Software Installation Steps

1. Installation for android/iOS

- Install node version greater than v16

```
brew install nvm - To install nvm first
```

```
nvm install 16 - To install specific node version
```

- Appium Installation:

- “npm install --global appium --drivers=xcuitest,uiautomator2” - To install Appium globally with required drivers for android and ios
- install apium doctor for health check “npm install appium-doctor -g” - To install appium doctor

- run “appium-doctor” in command prompt to check appium installation status and other dependency status

```
info AppiumDoctor Appium Doctor v.1.16.2
info AppiumDoctor ### Diagnostic for necessary dependencies starting ####
info AppiumDoctor ✓ The Node.js binary was found at: /Users/857851/.nvm/versions/node/v16.19.0/bin/node
info AppiumDoctor ✓ Node version is 16.19.0
info AppiumDoctor ✓ Xcode is installed at: /Applications/Xcode.app/Contents/Developer
info AppiumDoctor ✓ Xcode Command Line Tools are installed in: /Applications/Xcode.app/Contents/Developer
info AppiumDoctor ✓ DevToolsSecurity is enabled.
info AppiumDoctor ✓ The Authorization DB is set up properly.
WARN AppiumDoctor ✗ Carthage was NOT found!
info AppiumDoctor ✓ HOME is set to: /Users/857851
info AppiumDoctor ✓ ANDROID_HOME is set to: /Users/857851/Library/Android/sdk/
info AppiumDoctor ✓ JAVA_HOME is set to: /Library/Java/JavaVirtualMachines/adoptopenjdk-8.jdk/Contents/Home
info AppiumDoctor Checking adb, android, emulator, apk analyzer
info AppiumDoctor   'adb' is in /Users/857851/Library/Android/sdk/platform-tools/adb
info AppiumDoctor   'android' is in /Users/857851/Library/Android/sdk/tools/android
info AppiumDoctor   'emulator' is in /Users/857851/Library/Android/sdk/emulator/emulator
info AppiumDoctor   'apk analyzer' is in /Users/857851/Library/Android/sdk/cmake-tools/latest/bin/apk analyzer
info AppiumDoctor ✓ adb, android, emulator, apk analyzer exist: /Users/857851/Library/Android/sdk/
info AppiumDoctor ✓ 'bin' subfolder exists under '/Library/Java/JavaVirtualMachines/adoptopenjdk-8.jdk/Contents/Home'
info AppiumDoctor ### Diagnostic for necessary dependencies completed, one fix needed. #####
info AppiumDoctor
info AppiumDoctor ### Diagnostic for optional dependencies starting #####
WARN AppiumDoctor ✗ opencv4nodejs cannot be found.
WARN AppiumDoctor ✗ ffmpeg cannot be found.
WARN AppiumDoctor ✗ mjpeg-consumer cannot be found.
WARN AppiumDoctor ✗ set-simulator-location is not installed.
WARN AppiumDoctor ✗ idb and idb_companion are not installed.
WARN AppiumDoctor ✗ applesimutils cannot be found.
WARN AppiumDoctor ✗ ios-deploy cannot be found.
WARN AppiumDoctor ✗ bundletool.jar cannot be found.
WARN AppiumDoctor ✗ gst-launch-1.0 and/or gst-inspect-1.0 cannot be found.
info AppiumDoctor ### Diagnostic for optional dependencies completed, 9 fixes possible. #####
info AppiumDoctor
info AppiumDoctor ### Manual Fixes Needed #####
info AppiumDoctor The configuration cannot be automatically fixed, please do the following first:
WARN AppiumDoctor → [For lower than Appium 1.20.0] Please install Carthage. Visit https://github.com/Carthage/Carthage#installing-carthage for more information.
```

- Download and install appium desktop from below url

[Git.com](#)

Once installed move it to application folder and here you need admin permission.

- Configuring bash profile

- open terminal and run “vim ~/.bash_profile”
- Set ANDROID_HOME and JAVA_HOME path in bash profile
- Command to get JAVA_HOME “ls -l ‘which java’”
- Once you have both values add them in to bash profile as follow

```
▪ #ANDROID_HOME
```

```
export ANDROID_HOME=/Users/857851/Library/Android/sdk/
```

```
▪ #JAVA_HOME
```

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/adoptopenjdk-8.jdk/Contents/Home
```

- Cloning Automation Project from github

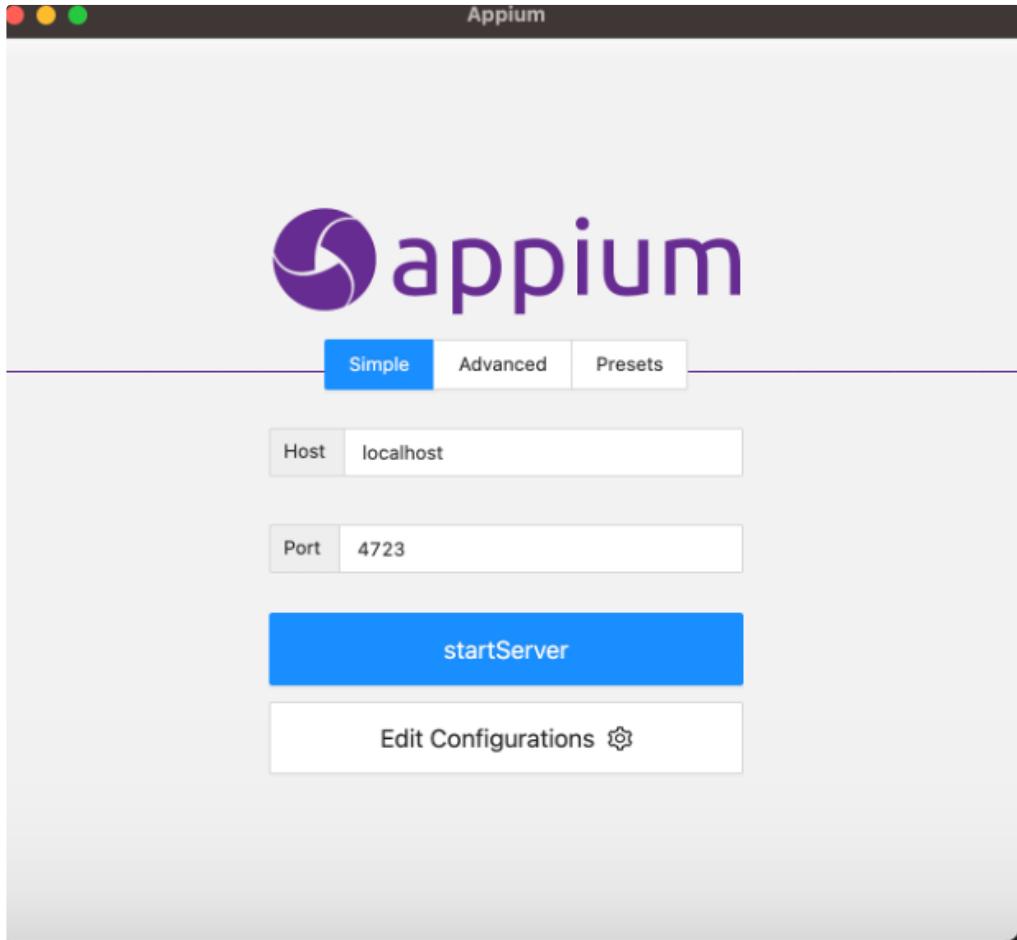
- Clone this repo “<https://github.com/victorycapital/vcm-ms-app-test-automation>” Connect your Github account - Connect your Github account ”

- Open the cloned project in editor and do “npm install”

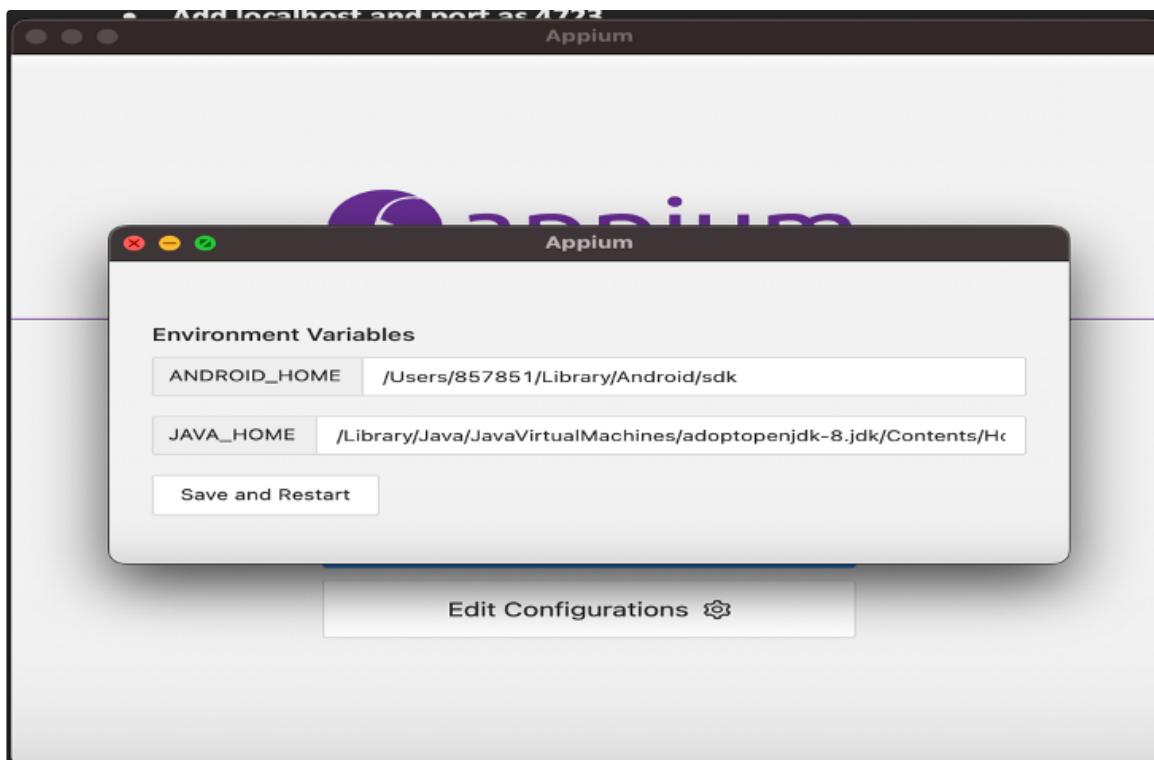
2. Configuring appium inspector

◦ Android Setup

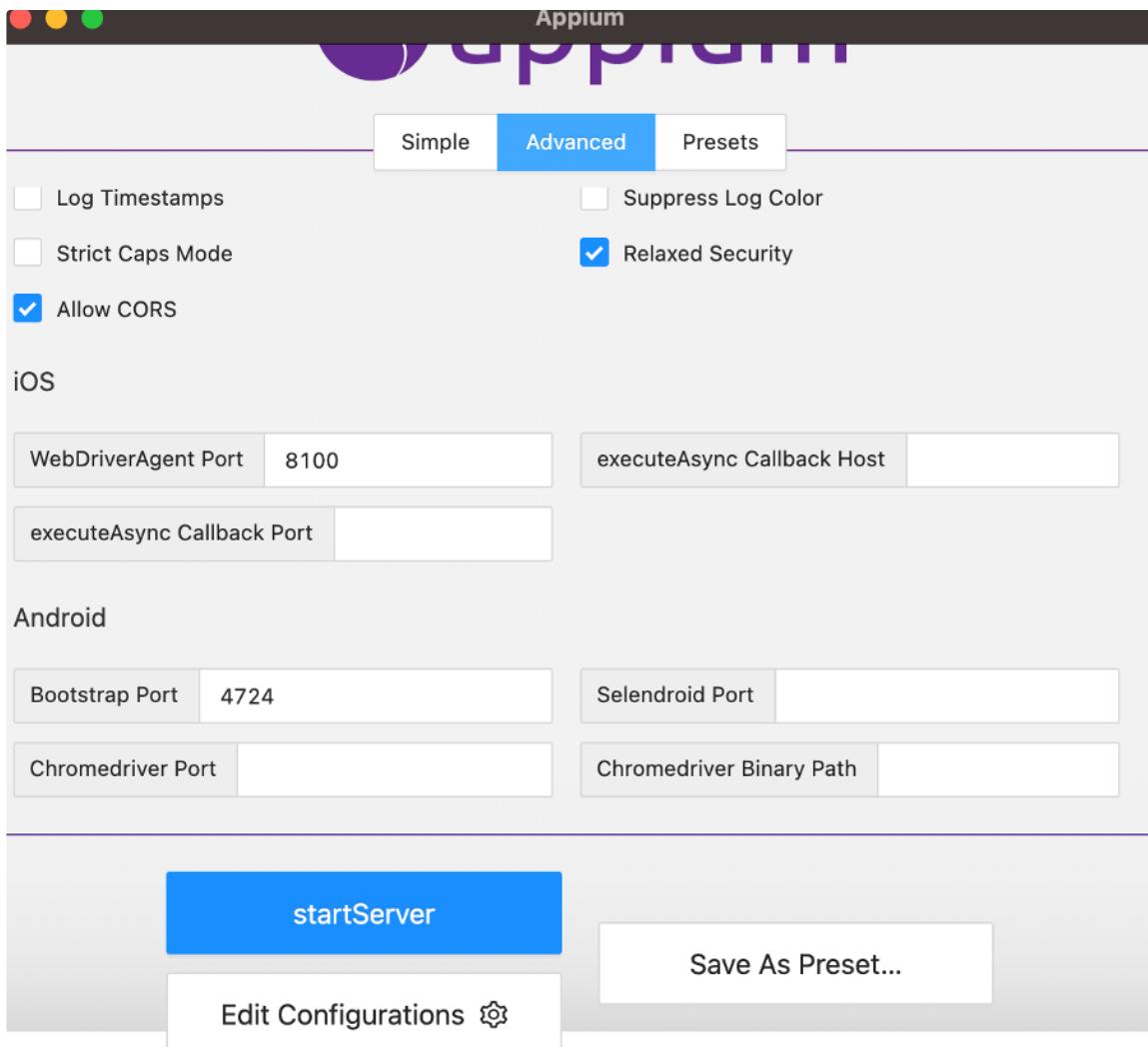
- i. Open the appium desktop app add hostname as “localhost” and port as “4723”



- ii. Now click on edit configuration and add android home path and java home path that we already added in bash profile

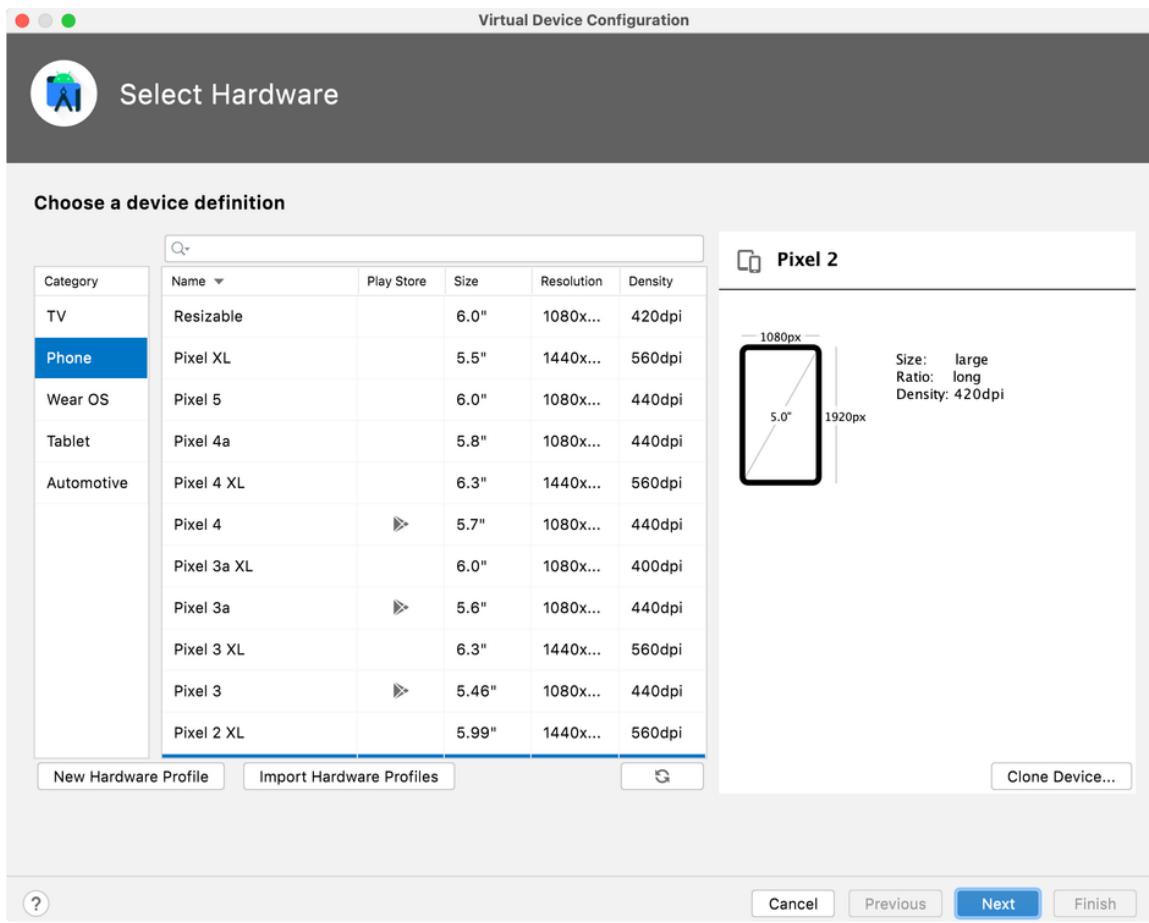


iii. Now click on advanced tab and select allow cors option and click start server

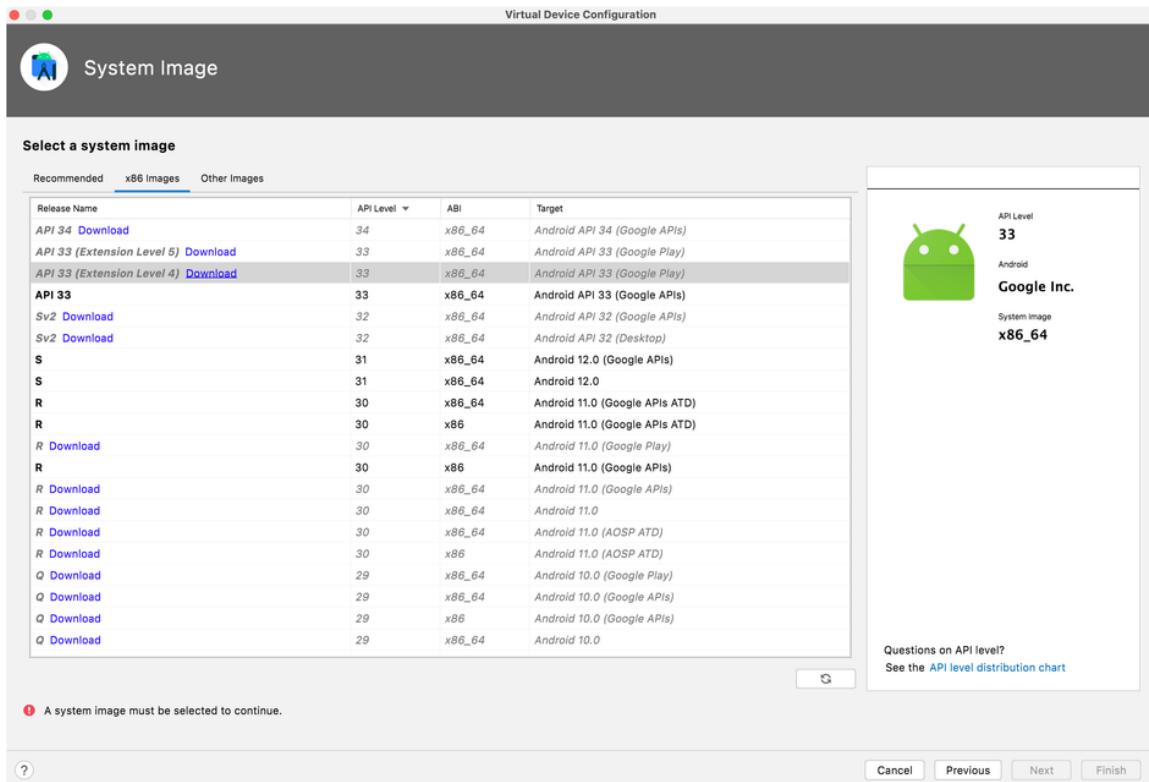


- **Configuring Android Simulator**

- Open Android studio and click on device manager in the top right
- Now click on create virtual device option and select the device you wish to create and then click on next



- Now in image selection screen click on x86 images and it will show the list available os to install
- Choose the OS that have google play option enabled



- Now click on next to install the selected os on the device
 - **Inspecting elements in android emulator**
 - Once you configured the emulator then open the browser and go to [Appium Inspector by Appium Pro site](#)
 - Now configure the below values in the site
- Remote Host - "localhost"**
- Remote Port - "4723"**
- Remote Path - "/wd/hub"**
- Now click on the Edit icon next to the capabilities and following values and save
- ```
"platformName": "Android",
"appium:platformVersion": "33",
"appium:automationName": "uiautomator2",
"appium:avd": "Nexus_5_API_33" → You can provide the device name that you created in android studio
```

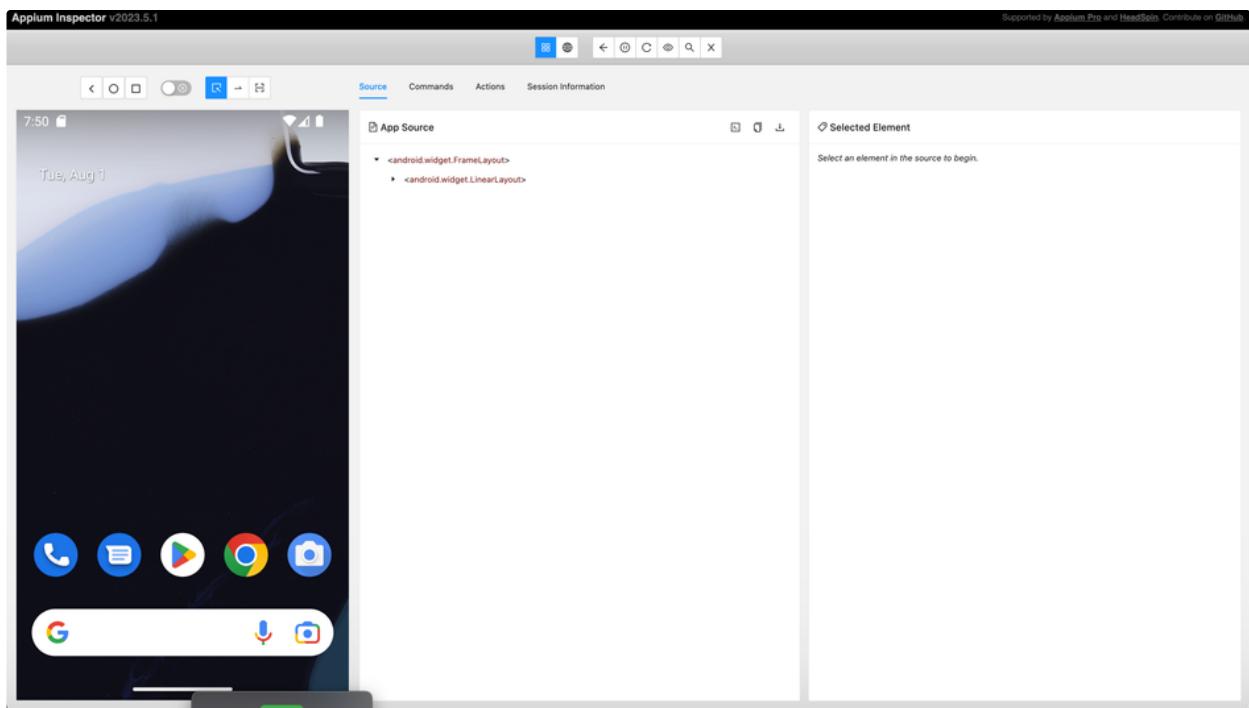
The screenshot shows the Appium Inspector interface with the following details:

- Appium Server:** Select Cloud Providers
- Remote Host:** localhost
- Remote Port:** 4723
- Remote Path:** /wd/hub
- SSL:** Unchecked
- Advanced Settings:** Expanded
- Desired Capabilities:**

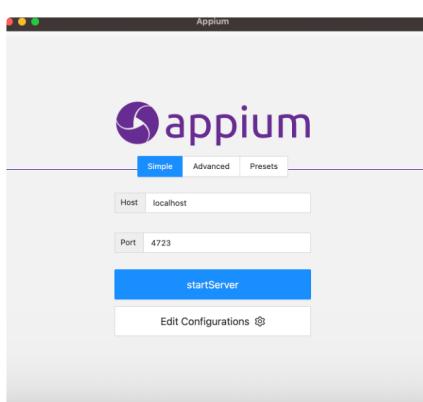
|                        | Value | Type           |
|------------------------|-------|----------------|
| platformName           | text  | Android        |
| appium:platformVersion | text  | 33             |
| appium:automationName  | text  | uiautomator2   |
| appium:avd             | text  | Nexus_5_API_33 |
- JSON Representation:**

```
{
 "platformName": "Android",
 "appium:platformVersion": "33",
 "appium:automationName": "uiautomator2",
 "appium:avd": "Nexus_5_API_33"
}
```
- Buttons at the bottom:**
  - Capabilities Documentation
  - Save As...
  - Start Session

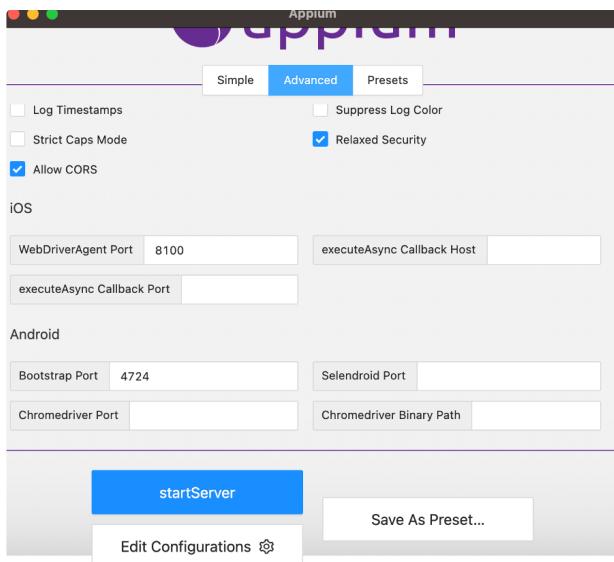
- Now click on start session and you will see the device screen in left



- Now drag and drop the application .apk file to the emulator to start inspect elements in the app
- **iOS Setup**
  - Open the appium desktop app add hostname as “localhost” and port as “4723”



- Now click on advanced tab and select allow cors option and click start server

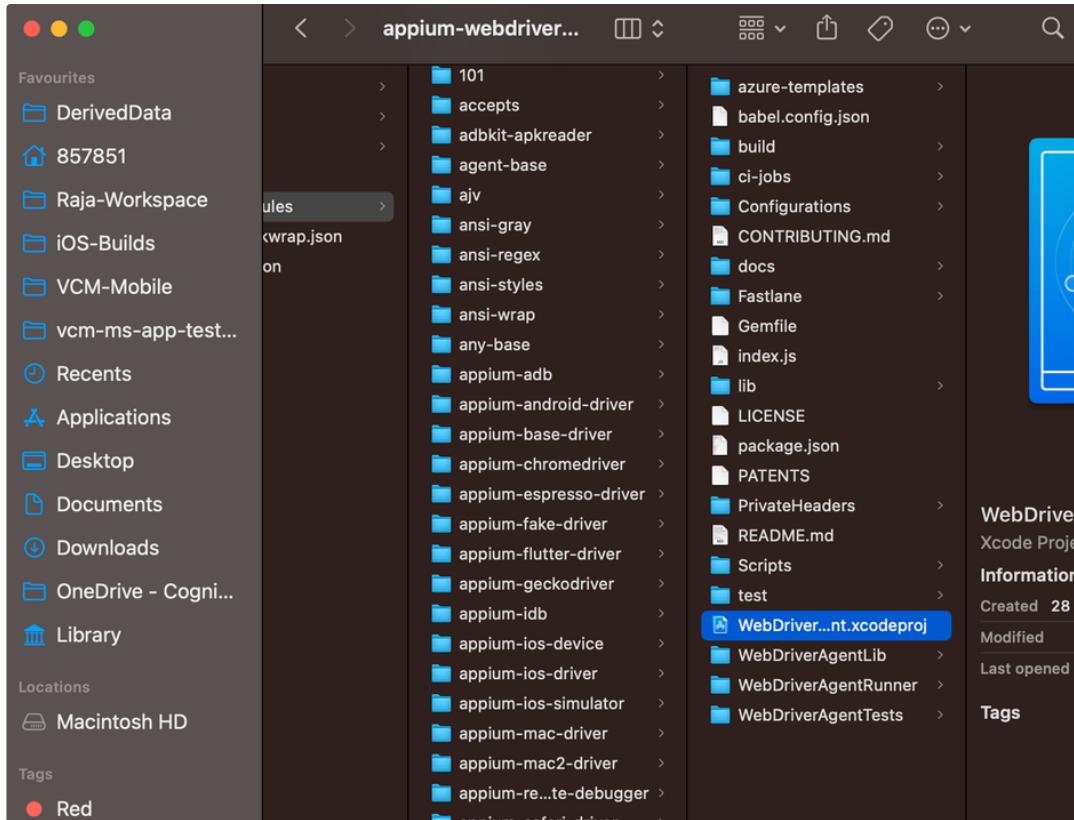


- o **Configuring iOS simulator**

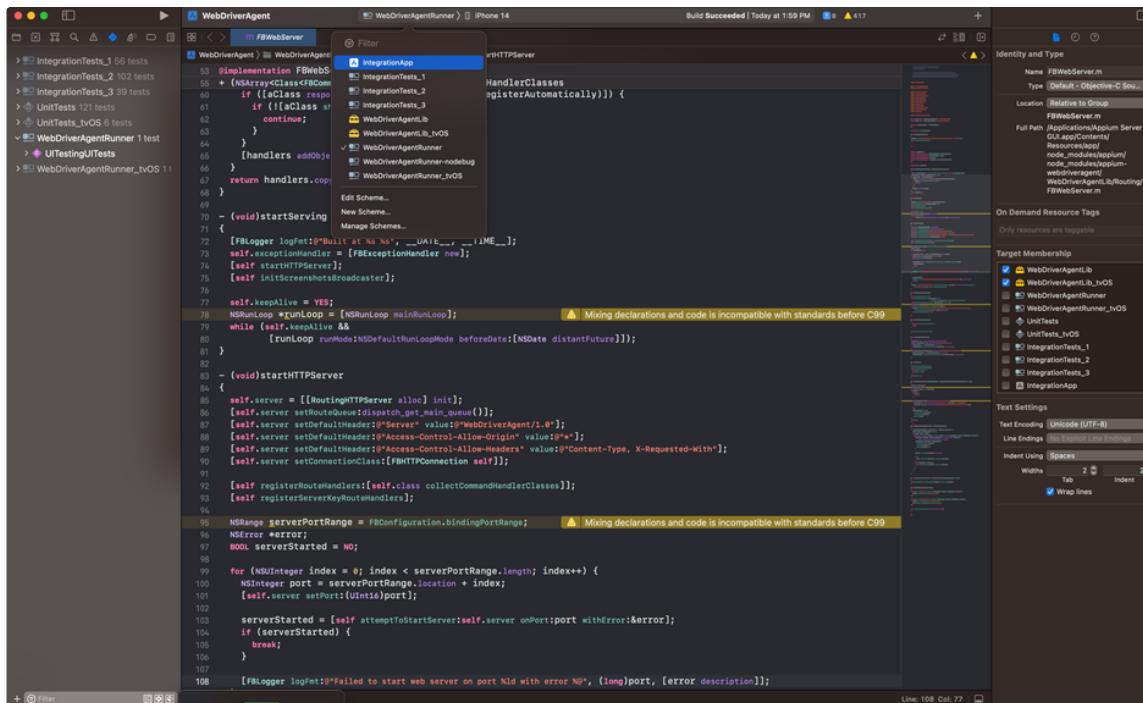
- Prerequisites(Admin access, apple developer account credentials)
- Go to the application folder in finder and click on appium desktop app
- Now right click on the app and click on show package contents



- Now got this path "Contents -> Resources → app → Node\_modules → Appium → Node\_modules → appium-webdriveragent → Open the Xcode project



- From the Xcode click on on target and choose webdriveragentRunner as below



- Now select the simulator you want to use for testing and then got to products tab and click on test options or click (cmd+U)
- Once the test has completed you will see the webdriveragent app installed in simulator.



- Now go to the inspector site [Appium Inspector by Appium Pro](#) and the following capabilities for iOS and click on start session.

A screenshot of the Appium Inspector interface. At the top, it says 'Appium Inspector v2023.5.1'. Below that, there are tabs for 'Appium Server' (which is selected) and 'Select Cloud Providers'. Under 'Appium Server', there are fields for 'Remote Host' (localhost), 'Remote Port' (4723), 'Remote Path' (/wd/hub), and an 'SSL' checkbox. There's also a link to 'Supported by Appium Pro and Headless'. In the main area, there's a section for 'Desired Capabilities' with tabs for 'Saved Capability Sets' and 'Attach to Session...'. Under 'Desired Capabilities', there are four rows of input fields:

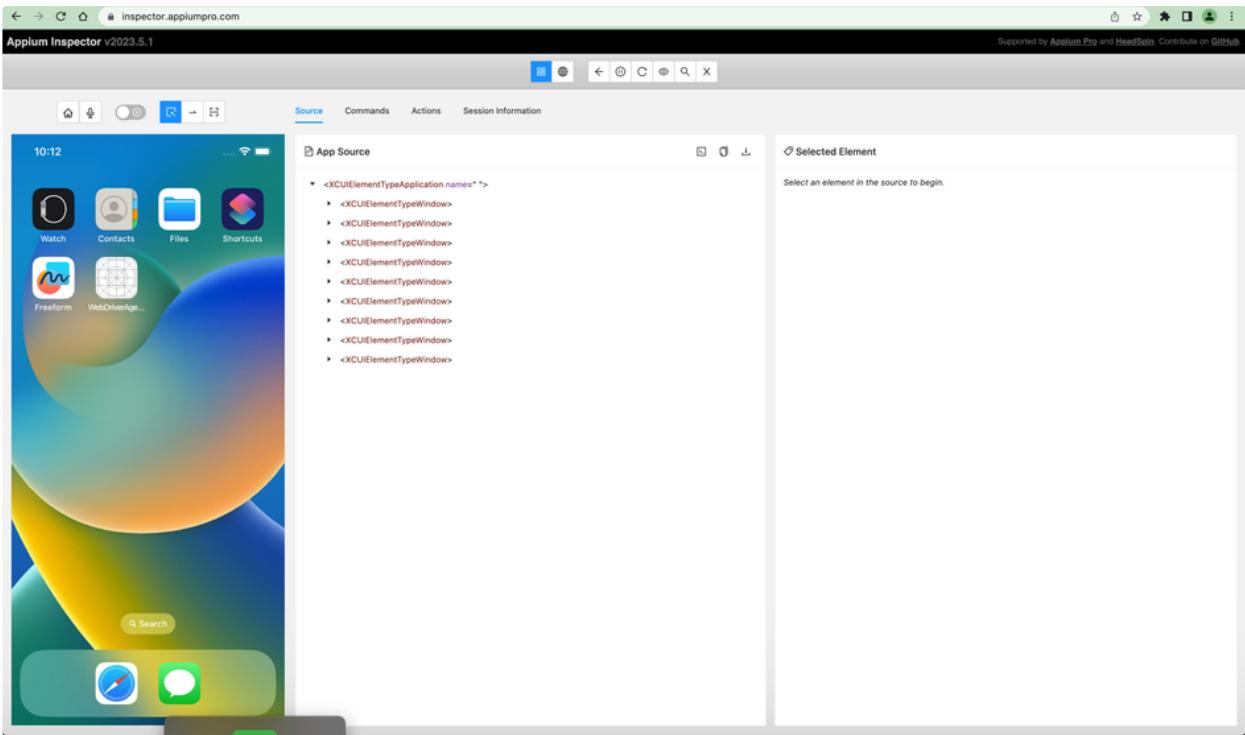
|                        |      |           |
|------------------------|------|-----------|
| platformName           | text | ios       |
| appium:platformVersion | text | 16.4      |
| appium:automationName  | text | XCUITest  |
| appium:deviceName      | text | iPhone 14 |

A checkbox labeled 'Automatically add necessary Appium vendor prefixes on start' is checked. To the right of these fields is a 'JSON Representation' panel containing the following JSON code:

```
{ "platformName": "ios", "appium:platformVersion": "16.4", "appium:automationName": "XCUITest", "appium:deviceName": "iPhone 14" }
```

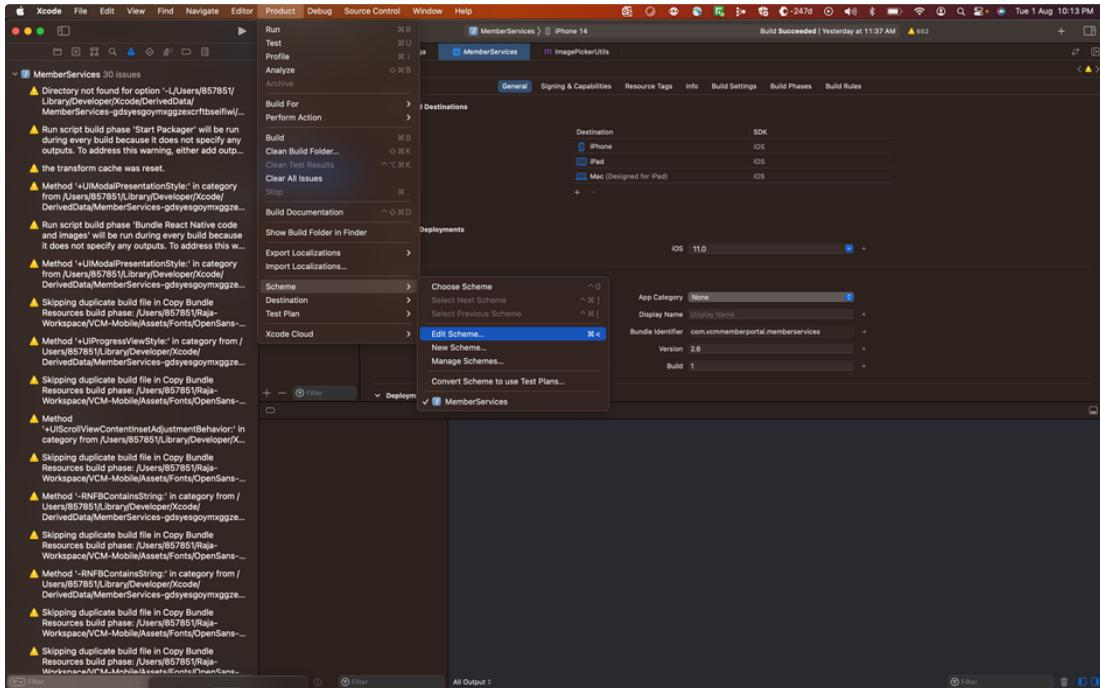
At the bottom of the interface, there are links for 'Capabilities Documentation', 'Save As...', and a large blue 'Start Session' button.

- Now you will see the simulator getting mirrored in the inspector.

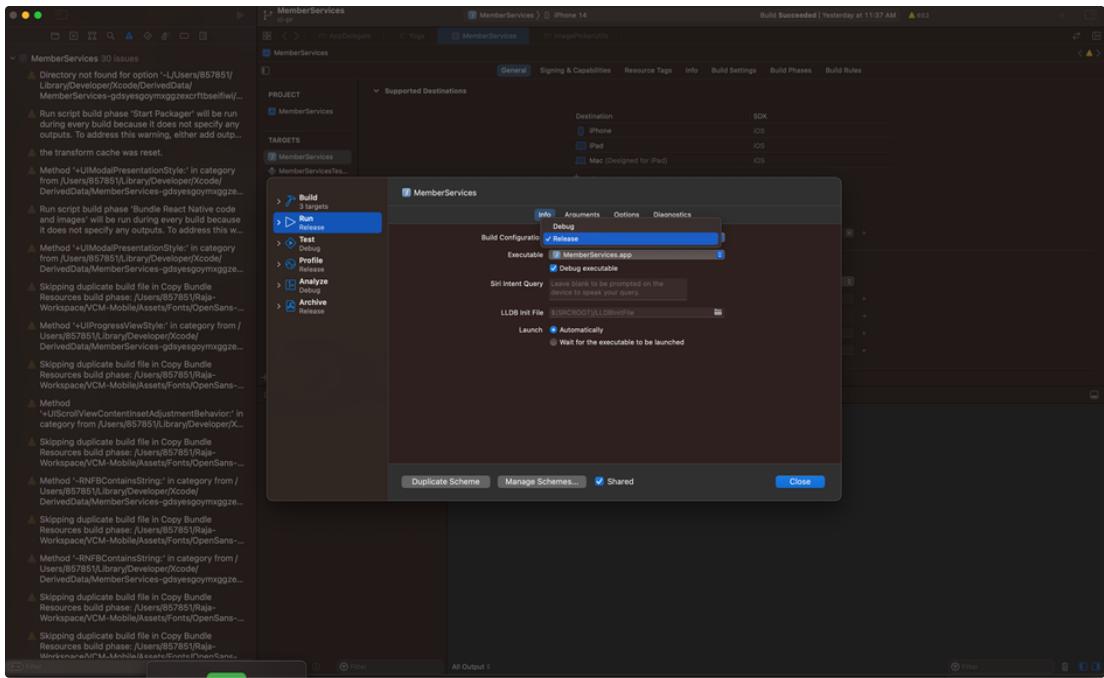


- Creating build for iOS

- For appium inspector .ipa file will not work so we need to create simulator build and following are the steps to create that.
- Open victory app in Xcode and go to "Products → Scheme → Edit Scheme"



- Now change the build configuration from debug to release and click on Products → Build or(cmd+B)



- Once build succeed go to products → show build folder in finder and you will be navigated to derived data
- Then go click on Build → Products → Release-iphonesimulator → MemberServices.app
- Drag and drop this to simulator and start inspect elements
- Add both .apk and this member services.app to our project in following path
  - vcm-ms-app-test-automation → apps → place both the builds here.