REQUIRED TO CHECK BEFORE WATCHING VIDEO!

What do I need for Appium installation on MacOS?

For all Mobile Testing subjects we will be using Appium, and main documentation for this is here: https://appium.io/docs/en/2.1/

Appium installation instructions for Mac OS users

Important Note: To see hidden files on Mac, press Command + Shift + . (dot)

1. At least JDK version 8 or a later version needs to be installed https://www.oracle.com/java/technologies/downloads/
2. The Java Development Kit is an implementation of either one of the Java Platform, Standard Edition, Java Platform, Enterprise Edition, or Java Platform, Micro Edition platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, macOS or Windows.

Hopefully, you have already installed JDK, open the terminal and run the command:

```
$ java --version
```

You should get something similar to:

```
java ~your-java-version~ 2018-07-17
Java(TM) SE Runtime Environment 18.3 (build 10.0.2+13)
Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.2+13, mixed mode)
```

Otherwise, proceed to this pageLinks to an external site. and look for a Mac OS X distribution, don't forget to accept the license agreement! Follow installation instructions and get java —version going.

3. Install NodeJS (latest LTS as of today) https://nodejs.org/en/

node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser.

npm is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command-line client, also called npm, and an online database of public and paid-for private packages called the npm registry.

Download here: https://nodejs.org/en/

Verify installation with node -v and npm -v.

```
$ node -v
```

$ npm -v

To give access for node_modules you can use following command if necessary (if you get EACCESS denied error) :

- sudo chown -R yourUserName:/usr/local/lib/

- Sudo is superuser can do anything.When you run this command it will ask you password and when you are typing it will be invisible.Type it correctly and click ENTER
- - Chown : Change Owner - To quit Node click Ctrl+C twice (if needed)

  Install Android Studio https://developer.android.com/studio
  (Simple installation - Always Next - Do not import Settings (if it asks) - For Data Sharing Don't Send - Install Type Standard - Finish-Choose SDK Manager)

Open Android Studio click on Configure and select AVD Manager:

Click at Create Virtual Device:

Proceed with Pixel 3 and hit Next:

Download Android 10 by clicking on Download link and following installation instructions:

Now it's time to create an emulator, select Android Q os image, and hit next:

Rename AND so the name doesn't contain spaces and click Next. Allow USB debugging

Run emulator by clicking play button:

4. Install Appium Server, you need to open your Terminal and type either (npm i appium -g) or (npm i --location=global appium)

Appium is an open-source test automation framework for use with native, hybridLinks to an external site., and mobile web apps. It drives iOS, Android, and Windows apps using the WebDriver protocol.

Most probably you won't have writing permissions to /usr/local/lib/node_modules (the place where npm will attempt to install Appium). To fix that you need to figure out the name of the current user:

```
$ whoami
```

For instance:

```
$ whoami
somedude
```

To allow for the current user to write to /usr/local/lib/node_modules run the command:

```
sudo chown -R $(whoami): /usr/local/lib/node_modules
sudo chown -R $(whoami): ~/.npm
```

Now you can run the npm command to install appium:

```
npm i appium -g
```

or

```
sudo npm i appium -g
```

IMPORTANT: Install UiAutomator2 Driver

Run This Command:

“appium driver install uiautomator2”

5. Install appium-doctor, you need to open your Terminal and type either of following (npm i appium-doctor -g) or (npm i --location=global appium-doctor)

To ensure proper appium setup let's install appium-doctor

```
npm install appium-doctor -g
```

or

```
sudo npm i appium-doctor -g
```

And execute setup verification with:

```
appium-doctor
...
WARN AppiumDoctor ✖ ANDROID_HOME is NOT set!
WARN AppiumDoctor ✖ JAVA_HOME is NOT set!
WARN AppiumDoctor  ✖ Carthage was NOT found!
...
```

At this point appium-doctor will complain about ANDROID_HOME, JAVA_HOME environment variables and Carthage, let's fix environment variables first. Create .zprofile (.bash_profile for older macs) file in the root:

```
touch ~/.zprofile
```

Open the file:

```
nano ~/.zprofile
```

And change the content of the .zprofile to:

```
export JAVA_HOME=$(/usr/libexec/java_home)
export ANDROID_HOME=/Users/$(whoami)/Library/Android/sdk
export PATH=$PATH:~/.local/bin$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools:$JAVA_HOME/bin
```

Example:

To save changes to the file hit Control + X, type y and hit Enter. Important, finish current terminal promt session (close all terminal windows) and start a new one, so changes to .bash_profile are in sync.

Alternatively you can open the folder with MAC User Interface -Go HOME

Important Note: To see hidden files on Mac, press Command + Shift + . (dot)

Click .zprofile to Open with text editor (Sublime etc)

Copy and Paste this command and save it with Command+S

```
export JAVA_HOME=$(/usr/libexec/java_home)

export ANDROID_HOME=/Users/{current-user-name}/Library/Android/sdk

export PATH=$PATH:~/.local/bin$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools:$JAVA_HOME/bin
```

6. Install Appium Inspector https://github.com/appium/appium-inspector/releases/tag/v2021.9.2Links to an external site.

7. Install carthage using homebrew:

https://github.com/Carthage/Carthage#installing-carthage

Homebrew: You can use Homebrew and install the `carthage` tool on your system simply by running `brew update` and `brew install carthage`.

https://brew.sh/

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Then install carthage

```
$ brew install carthage
```

Run appium-doctor again, everything should be green now.

8. Create Android Emulator (Pixel 3 Android 10)

9. Create a basic Maven project

10. Add dependencies and create basic test class:

```
<dependencies>
    <!-- https://mvnrepository.com/artifact/io.appium/java-client -->
    <dependency>
        <groupId>io.appium</groupId>
        <artifactId>java-client</artifactId>
        <version>7.6.0</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api -->
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-api</artifactId>
        <version>5.8.2</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

```
import io.appium.java_client.AppiumDriver;
import io.appium.java_client.MobileElement;
import io.appium.java_client.android.AndroidDriver;
import io.appium.java_client.remote.MobileCapabilityType;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.remote.DesiredCapabilities;

import java.net.MalformedURLException;
import java.net.URL;

public class FirstAppiumTest {
    AppiumDriver<MobileElement> driver;

    @Test
    public void test1() throws MalformedURLException, InterruptedException {
        DesiredCapabilities capabilities = new DesiredCapabilities();
        capabilities.setCapability("deviceName", "Pixel 3");
        capabilities.setCapability(MobileCapabilityType.PLATFORM_NAME, "Android");
        capabilities.setCapability(MobileCapabilityType.PLATFORM_VERSION, "10");
        capabilities.setCapability(MobileCapabilityType.AUTOMATION_NAME,
"UiAutomator2");
        capabilities.setCapability(MobileCapabilityType.APP, "https://cybertek-
appium.s3.amazonaws.com/calculator.apk");

        URL url = new URL("http://localhost:4723/");

        driver = new AndroidDriver<MobileElement>(url, capabilities);
        Thread.sleep(3000);
        driver.closeApp();
    }
}
```