Important notes:
- Use node LTS and not current version
- Use Appium latest and not some older version
- Be admin on your MAC
- If on corpnet (office laptop/computer), make sure anti-virus is not blocking node and Appium installation. Work with your security team in case of issues

Install homebrew [package manager for macOS and is used to install software packages]
==========
Link: https://brew.sh/
Command: /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

Install node and npm [Appium dependencies]
==========
Commands to check if node and npm are installed:
node -v
npm -v
Command to install node: brew install node [This will install npm as well]
Command to check node installation path: where node or which node

Install Appium server using NPM (Appium CLI)
==========
Command to install Appium: npm install -g appium
Command to check Appium version: appium -v
Command to check Appium installation path: where appium or which Appium

Install Appium server using Appium Desktop client
==========
Download link: https://appium.io

Install Appium Inspector
==========
-> Download and install from https://github.com/appium/appium-inspector/releases

Install Xcode
==========
Configure Apple ID in Account preferences

Install from App Store

Install Xcode command line tools
==========
Command: xcode-select --install

Install xcpretty [to make Xcode output reasonable]
==========
Command to install xcpretty: gem install xcpretty

Install Carthage [dependency manager, required for WebDriverAgent]
==========
Command to install Carthage: brew install Carthage

Install Appium-doctor and check Appium setup
==========
Command to install Appium doctor: npm install -g appium-doctor
Command to get help: appium-doctor --h
Command to check setup for iOS: appium-doctor --bios

Command to get UDID
==========
Command to get UDID: xcrun simctl list
Command to get UDID: xcrun xctrace list devices
XCode option to get UDID: XCode -> Window -> Devices and Simulators -> Simulators -> Select the simulator in the left pane. In the right pane, "Identifier" will be the UDID.

==========
Real device setup
==========

Getting UDID
==========
Command to install ios-deploy: npm install -g ios-deploy
Command to get UDID: ios-deploy -c
OR
Command to get UDID: xcrun simctl list [May show real devices as well]
Command to get UDID: xcrun xctrace list devices [May show real devices as well]
XCode option to get UDID: XCode -> Window -> Devices and Simulators -> Devices -> Select the device in the left pane. In the right pane, "Identifier" will be the UDID.

1. Code signing WebDriverAgent: Basic (automatic/manual) configuration.
==========

Step 1. Enroll for Developer program
- Create Apple account: https://developer.apple.com
- Enable two factor authentication: https://appleid.apple.com/account/manage
- Click Join the Apple Developer program
- Click Enroll
- Click Start Your Enrollment

Step 2. Register device UDID on the developer portal (this can be done from Xcode as well)

Step 3. Add your Apple ID (paid developer account) to XCode and download the certificate (if required, create new certificate on the developer portal)

2. Commands to get the Bundle ID of already installed app on your real device
==========

```
# For OSX
brew install libimobiledevice
brew install ideviceinstaller
# For Ubuntu
sudo apt-get install libimobiledevice
sudo apt-get install ideviceinstaller
# Then plugin your device and run the following to get your app installed:
ideviceinstaller -l
```

You can watch the full video from here (https://www.youtube.com/watch?v=T4iTJUjB1iE)
Another approach to get the bundle id from iTunes (https://www.youtube.com/watch?v=kigm5Erw4tI&t=91s)

3. Following are the Instructions to run the application on Real iPhone/Simulator Device
==========

Installing Webdriver Agent on your real device

1 Make sure the real device is connected
2.Navigate to applications folder on your mac
3 Right click on Appium Server GUI and click on show package contents

4. Navigate to the following path - /Applications/Appium\ Server\ GUI.app/Contents/Resources/app/node_modules/appium/node_modules/appium-webdriveragent (An easy way to go to this path is to search for .xcodeproj and open the WebDriverAgentMac.xcodeproj in Xcode)
5. Open WebDriverAgent.xcodeproj in Xcode.
6. Perform the following steps for WebDriverAgentLib, Integration-app and WebDriverAgentRunner targets

* Select WebDriverAgent Agent Lib > Select "Automatically manage signing" in the "Signing & Capabilities" tab, and then select your Team (You should be logged in with your paid apple development account to see your Team in Xcode)
* On the menubar, select WebDriverAgentLib> Your real device and then Product > build.

Perform the * steps under point 6 for Integration-app and WebDriverAgentRunner targets

7 Finally You can click on Product > Test (It will install webdriver agent runner on your real device)

Use the following mandatory capabilities to launch the already installed app on iPhone Real/Simulator Device on appium Inspector
==========

```
{
  "platformName": "iOS",
  "appium:deviceName": "iPhone 11 Pro",
  "appium:automationName": "XCUITest",
  "appium:udid": "<Your device's UDID>",
   "xcodeOrgId": "<Team ID>",
   "xcodeSigningId": "iPhone Developer",
  "appium:bundleId": "com.apple.mobilecal", // This is the bundle Id of calendar application
  "appium:platformVersion": "15.0"
}
```

Rory, you can skip the Android Steps if you don't want to run the application on the Android Device(s)

Following are the Instructions to run the application on Android Real/ Emulator Device
==========

Install Android Studio
==========
- Android Sutdio download link: https://developer.android.com/studio

Set JAVA_HOME and ANDROID_HOME environment variables
==========
Option1 (zprofile - MacOS Catalina default shell is zsh):

-> Navigate to home directory: cd ~/
-> Open zprofile file: open -e .zprofile
-> Create zprofile file: touch .zprofile
-> Add below entries:
export JAVA_HOME=$(/usr/libexec/java_home)
Important note: If above path doesn't work, try /Library/Java/
JavaVirtualMachines/your_jdk_version/Contents/Home
Here, your_jdk_version can be jdk15.0.2.jdk for example.
export ANDROID_HOME=${HOME}/Library/Android/sdk
export PATH="${JAVA_HOME}/bin:${ANDROID_HOME}/tools:${ANDROID_HOME}/
platform-tools:${PATH}"
-> source .zprofile

Option2 (zprofile and bashprofile):

-> Navigate to home directory: cd ~/
-> Open bash profile file: open -e .bash_profile
-> Create bash profile: touch .bash_profile
-> Add below entries:
export JAVA_HOME=$(/usr/libexec/java_home)
Important note: If above path doesn't work, try /Library/Java/
JavaVirtualMachines/your_jdk_version/Contents/Home
Here, your_jdk_version can be jdk15.0.2.jdk for example.
export ANDROID_HOME=${HOME}/Library/Android/sdk
export PATH="${JAVA_HOME}/bin:${ANDROID_HOME}/tools:${ANDROID_HOME}/
platform-tools:${PATH}"
-> Open zprofile file: open -e .zprofile
-> Create zprofile file: touch .zprofile
-> add this line: source .bash_profile
echo $JAVA_HOME
echo $ANDROID_HOME
echo $PATH

Verify installation using appium-doctor (Used to check if the IOS and
Android setup is done properly)
==========
- Command to install appium-doctor: npm install -g appium-doctor
- Command to get appium-doctor help: appium-doctor --help
- Command to check Android setup: appium-doctor --android

Emulator Setup: Create AVD and start it
==========
Open Android Studio
Click Configure option

Click "AVD Manager" option
Click "Create Virtual Device" button
Select the phone model
Download the Image for desired OS version if not already downloaded
Start AVD

Emulator Setup: Create driver session with the AVD using Appium Inspector
==========
Download link for dummy app:
https://github.com/appium/appium/blob/master/sample-code/apps/ApiDemos-debug.apk

Note: If using Appium desktop, might get error with adb tool because Appium Desktop
cannot read ANDROID_HOME and JAVA_HOME path from the zsh/bash profile.
To resolve, set ANDROID_HOME to SDK path and JAVA_HOME to Java home path using "Edit Configurations" option
while launching the Appium Desktop.

Real Device Setup (Android): Enable USB debugging on Android mobile
==========
On your phone,
Go to Settings
Click System option
Click "About Phone" option
Click on "Build Number" 7 to 8 times
Go back to Settings
Open Developer Options
Enable "USB Debugging"

Real Device Setup: Create driver session using Appium Inspector
==========
Download link for dummy app:
https://github.com/appium/appium/blob/master/sample-code/apps/ApiDemos-debug.apk

Note: If using Appium desktop, might get error with adb tool because Appium Desktop
cannot read ANDROID_HOME path from the zsh/bash profile.
To resolve, set ANDROID_HOME to SDK path using "Edit Configurations" option
while launching the Appium Desktop.

Commands to get the appPackage and appActivity on Android Devices
==========

Make sure the app is Install on device and launch the application on your device and run the following command

```
adb shell "dumpsys activity activities | grep mResumedActvity"
```

Other easy way is install the application apk info on android device and it will list the app package and app activity of all the installed apps

Use the following mandatory capabilities to launch the already installed app on Android Emulator on appium Inspector

```
{
  "platformName": "Android",
  "appium:deviceName": "Emulator", // you can use any random name
  "appium:automationName": "UiAutomator2",
  "appium:udid": "emulator-5554", // run the adv devices to get the UDID of your real/emulator android device
  "appium:appPackage": "com.apple.mobilecal", // to launch calendar app
  "appium:appActivity": "com.apple.mobilecal.SplashScreenActivity"
}
```