RBCN 2021 REMOTE EVENT

MAR 2021

#ROBOCON20

Robot Framework Foundation

Getting Started With The Robotframework-AppiumLibrary

:: RoboCon Workshop Presented By ::

Joshua Gorospe
Principal QA Engineer
Secureworks https://www.secureworks.com/

Linkedin Profile https://www.linkedin.com/in/joshua8481/
Twitter Profile https://twitter.com/8481jg

ROBOCON 2021

Before We Begin...

I want to thank the following people and groups.

- Pekka Klarck
- Ed Manlove
- Serhat Bolsu
- Antti Karjalainen
- René Rohner
- James Bach
- Jonathan Bach
- Michael Bolton
- Cem Kaner
- Kristian Karl
- Harry Robinson
- Mikko Korpela
- Bryan Oakley
- Shiva Prasad Adirala
- The Appium development team
- The Graphwalker development team
- The thought-leaders who inspired me to learn model-based testing
- The thought-leaders of Rapid Software Testing Methodology
- The entire Robot Framework community and its contributors

FYI Windows Users

The Robotframework-AppiumLibrary examples in this workshop use a lot of Bash scripting. Here are some possible options that may help you, but I have not tested them myself...

- Windows Subsystem for Linux https://docs.microsoft.com/en-us/windows/wsl/
- Git for Windows https://git-scm.com/download/win
- CygWin https://www.cygwin.com/


MAR 2021

Introduction And Workshop Outline

This will be a very hands on Workshop covering Robotframework-AppiumLibrary combined with various tools. I will be frequently moving back and forth between slides and running the examples from a terminal. From a high-level, the following Workshop areas will be covered in two parts.

- Part One
  - Technical requirements summary.
  - Appium installation advice and mobile device Emulator/Simulator setup guidance.
  - App automation workflows for IOS, and Android (including adb shell).
  - Browser automation workflows for Safari on IOS and Chrome on Android.

- Part Two
  - Appium Desktop walkthrough, and useful features for Robot Framework users.
  - Charles Proxy combined with Robotframework-AppiumLibrary.
  - PaBot running IOS + Android simultaneously, combined with Graphwalker, and Android Emulator CPU monitoring.

Part One: Technical Requirements Summary

The following are needed to run the Workshop Examples. Please follow instructions on each website to install them.

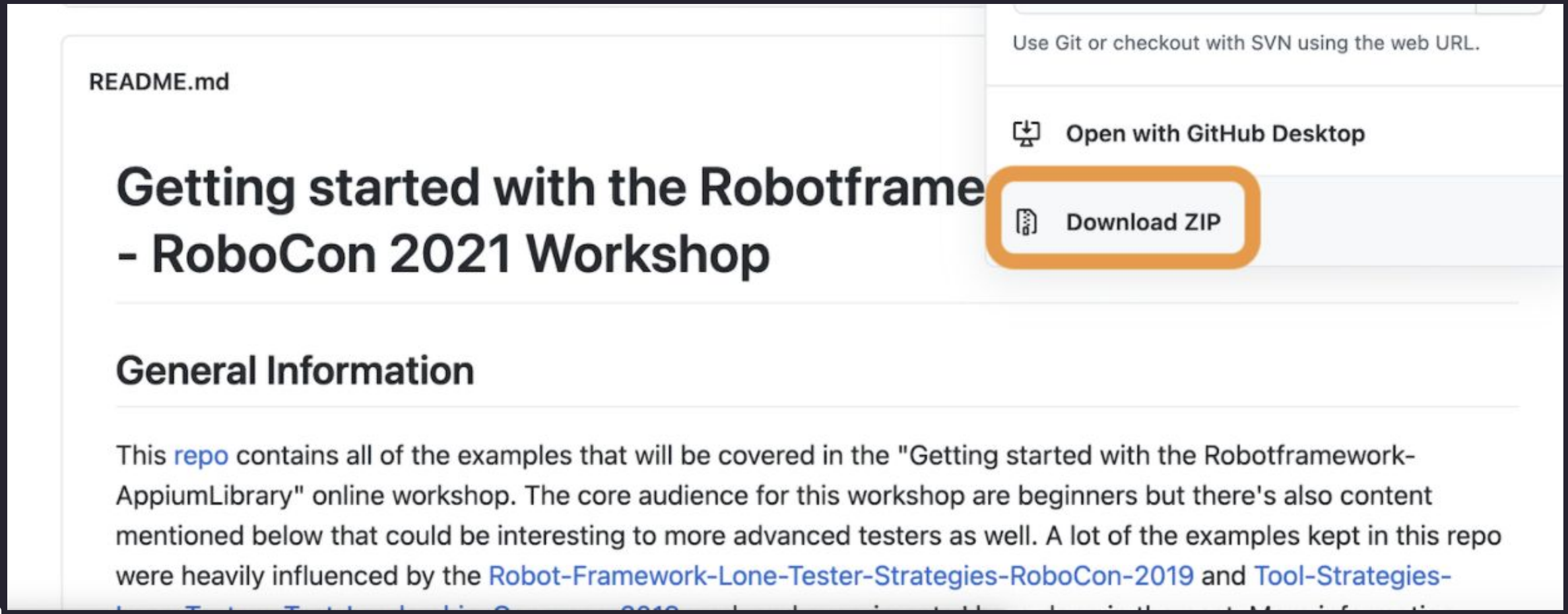
- Python 3, this workshop was created using version 3.9.2 -> https://www.python.org/downloads/
- NodeJS 14, this workshop was created using version 14.16.0 -> https://nodejs.org/en/ or https://github.com/nvm-sh/nvm
- Java 11, this workshop was created using version 11.0.10 -> https://www.java.com/en/
- Appium Server -> http://appium.io/docs/en/about-appium/getting-started/?lang=en#getting-started
- Android Studio -> https://developer.android.com/studio
- XCode, only available for MacOS users -> https://developer.apple.com/xcode/
- Robot Framework and Robotframework-AppiumLibrary -> https://robotframework.org
- PaBot -> https://github.com/mkorpela/pabot
- Charles Proxy -> https://www.charlesproxy.com/documentation/proxying
- Graphwalker -> https://github.com/GraphWalker/graphwalker-project
- Appium Desktop -> https://github.com/appium/appium-desktop
- wget -> https://www.gnu.org/software/wget/
- cURL -> https://curl.se/
- jq https://stedolan.github.io/jq/
- git https://git-scm.com/

MAR 2021

Part One: Technical Requirements Summary

Run the following commands to get the Workshop Examples (requires git).

- git clone https://github.com/jg8481/Getting-Started-Robotframework-AppiumLibrary-RoboCon-2021.git
- cd ./Getting-Started-Robotframework-AppiumLibrary-RoboCon-2021
- You can also download a zip file from the GitHub page -> https://github.com/jg8481/Getting-Started-Robotframework-AppiumLibrary-RoboCon-2021

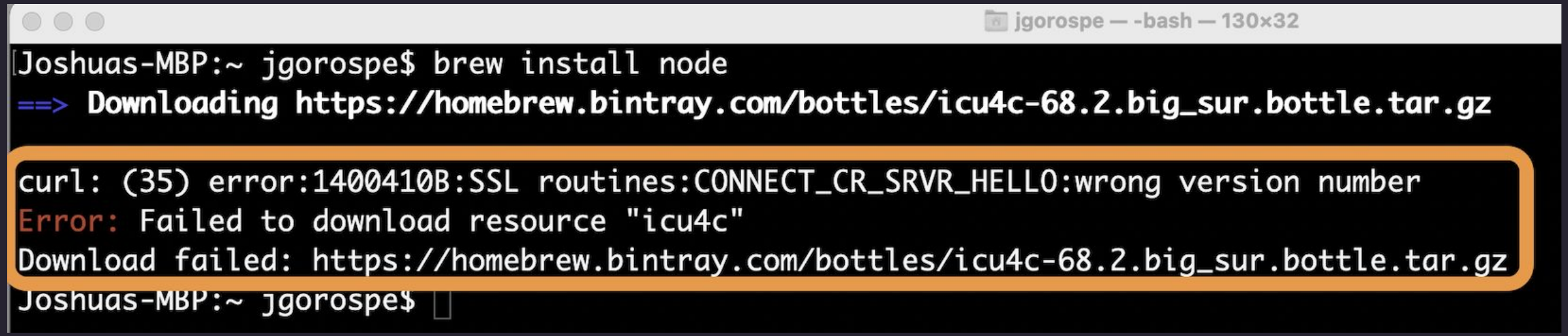


MAR 2021

Part One: Appium Installation Advice, Android Emulator, IOS Simulator Setups

Please make sure you have reviewed all of the items in the Technical Requirements Summary slides.

- The rest of this Part One section assumes you have installed all of the basics.
- Basic requirements for this Workshop are NodeJS 14, Python 3, and Java 11
  - I recommend NVM (https://github.com/nvm-sh/nvm) for NodeJS installation.
  - Installing anything with Homebrew (https://brew.sh/) may fail sometimes, for example this happened after I installed the MacOS Big Sur update.

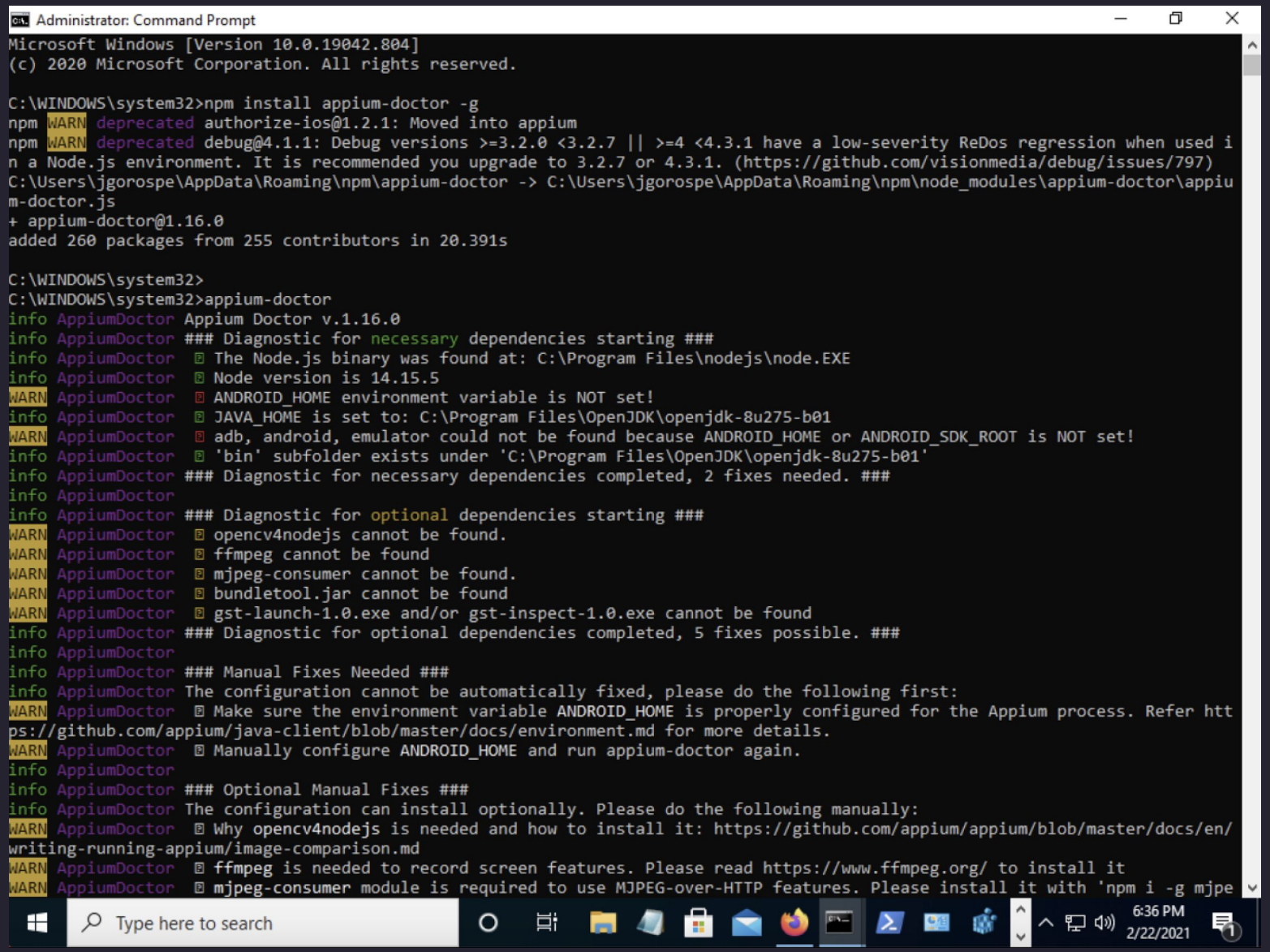


Based on the Appium "Getting Started" instructions kept here -> http://appium.io/docs/en/about-appium/getting-started/index.html, you should be able to run the following commands successfully without issues.

- Install Appium Server -> npm install -g appium
- Install appium-doctor -> npm install -g appium-doctor
- Run appium-doctor...

Part One: Appium Installation Advice, Android Emulator, IOS Simulator Setups

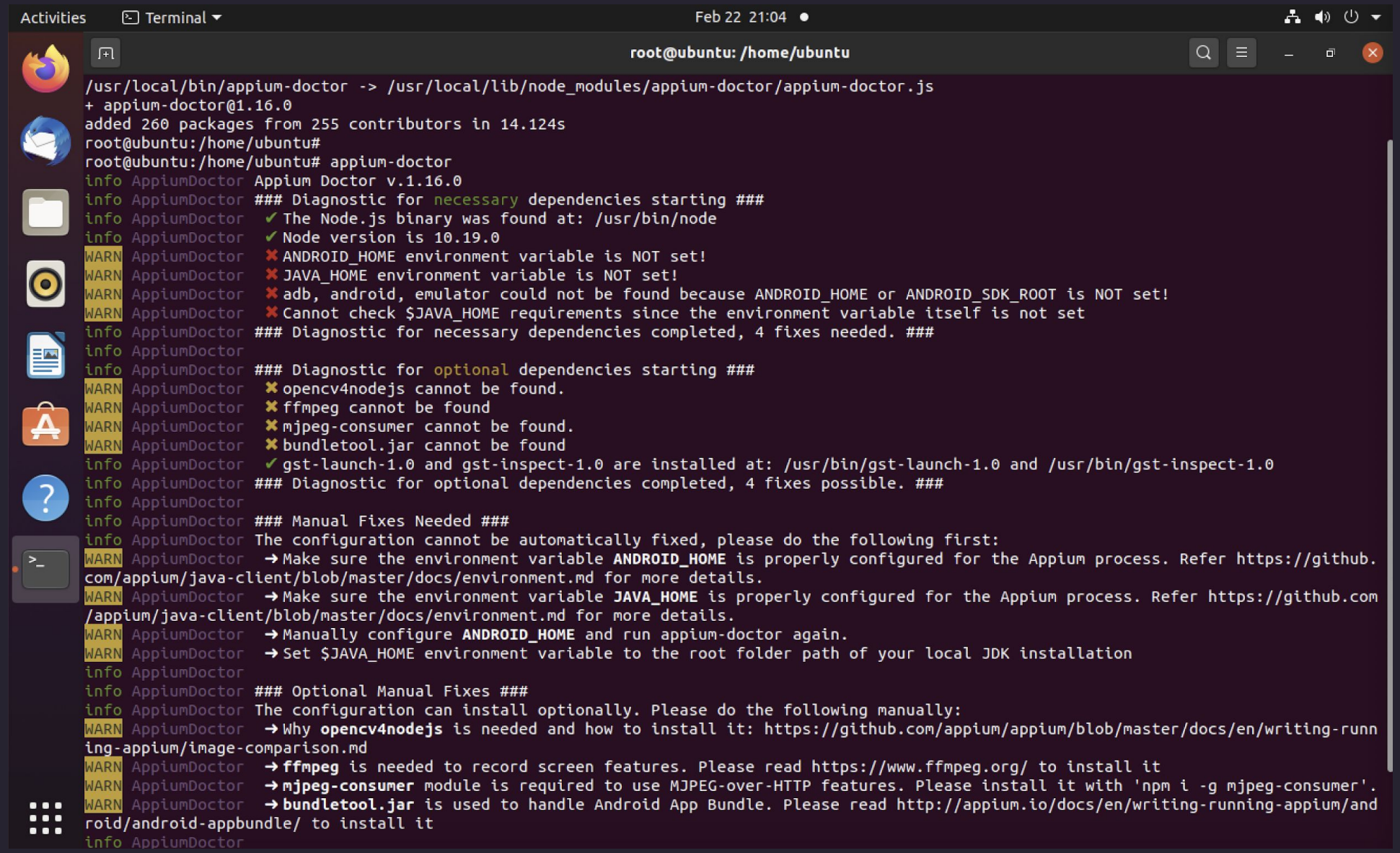
From a Windows operating system the appium-doctor output should look similar to this.



MAR 2021

Part One: Appium Installation Advice, Android Emulator, IOS Simulator Setups

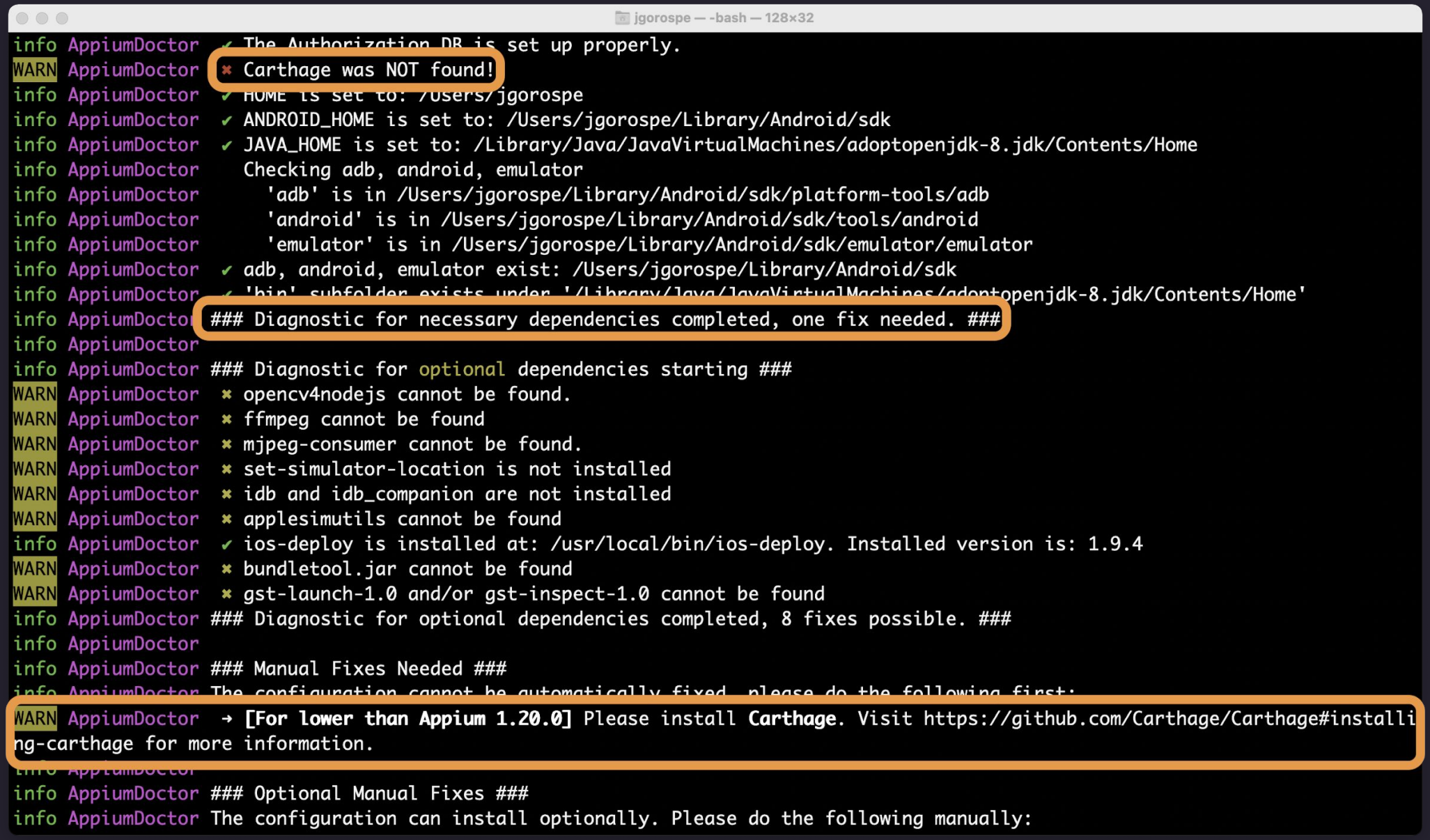
From a Linux (Ubuntu) operating system the appium-doctor output should look similar to this.



MAR 2021

Part One: Appium Installation Advice, Android Emulator, IOS Simulator Setups

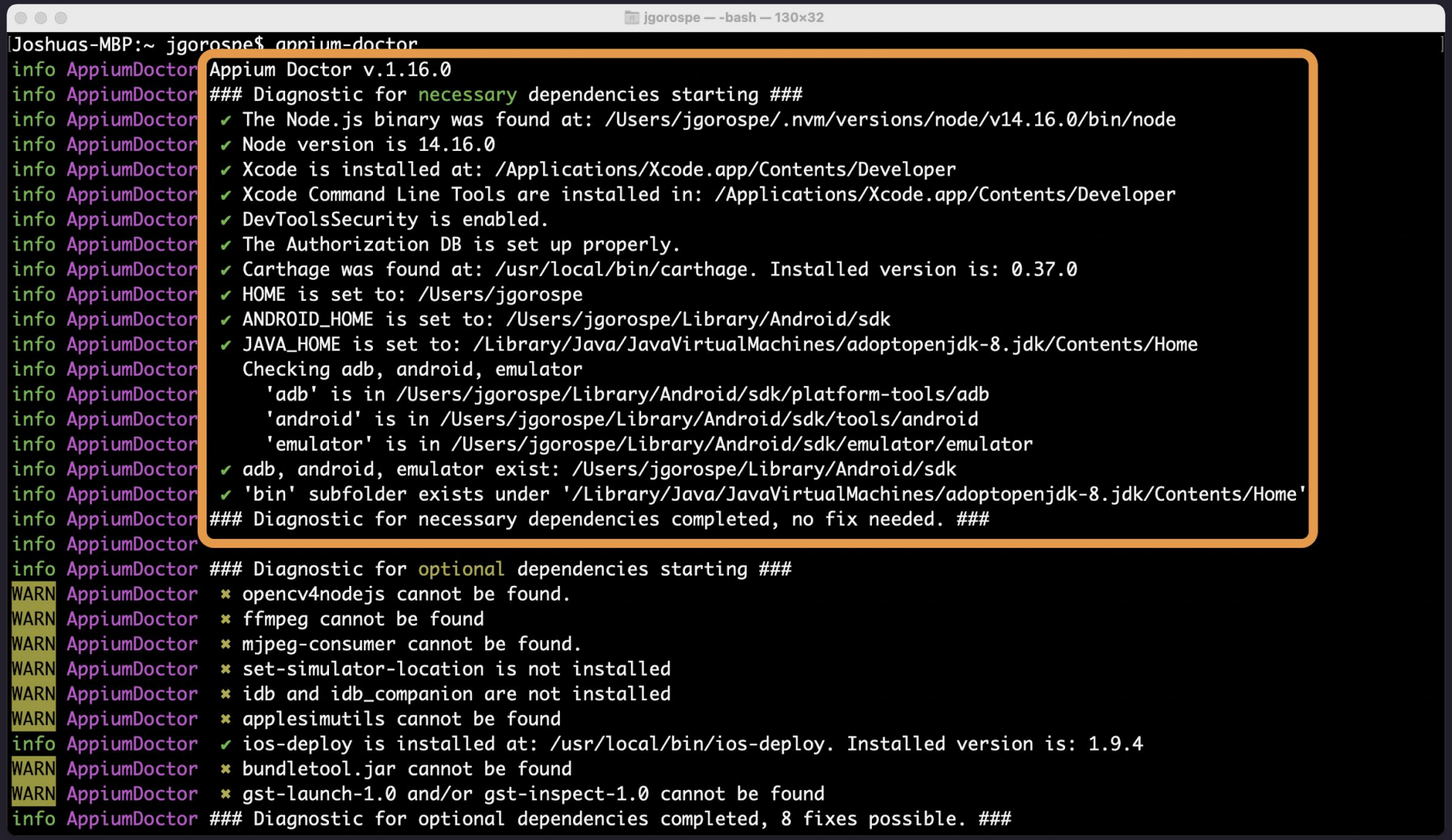
Pay attention to any red "X" output. Follow all of the instructions provided by appium-doctor.



MAR 2021

Part One: Appium Installation Advice, Android Emulator, IOS Simulator Setups

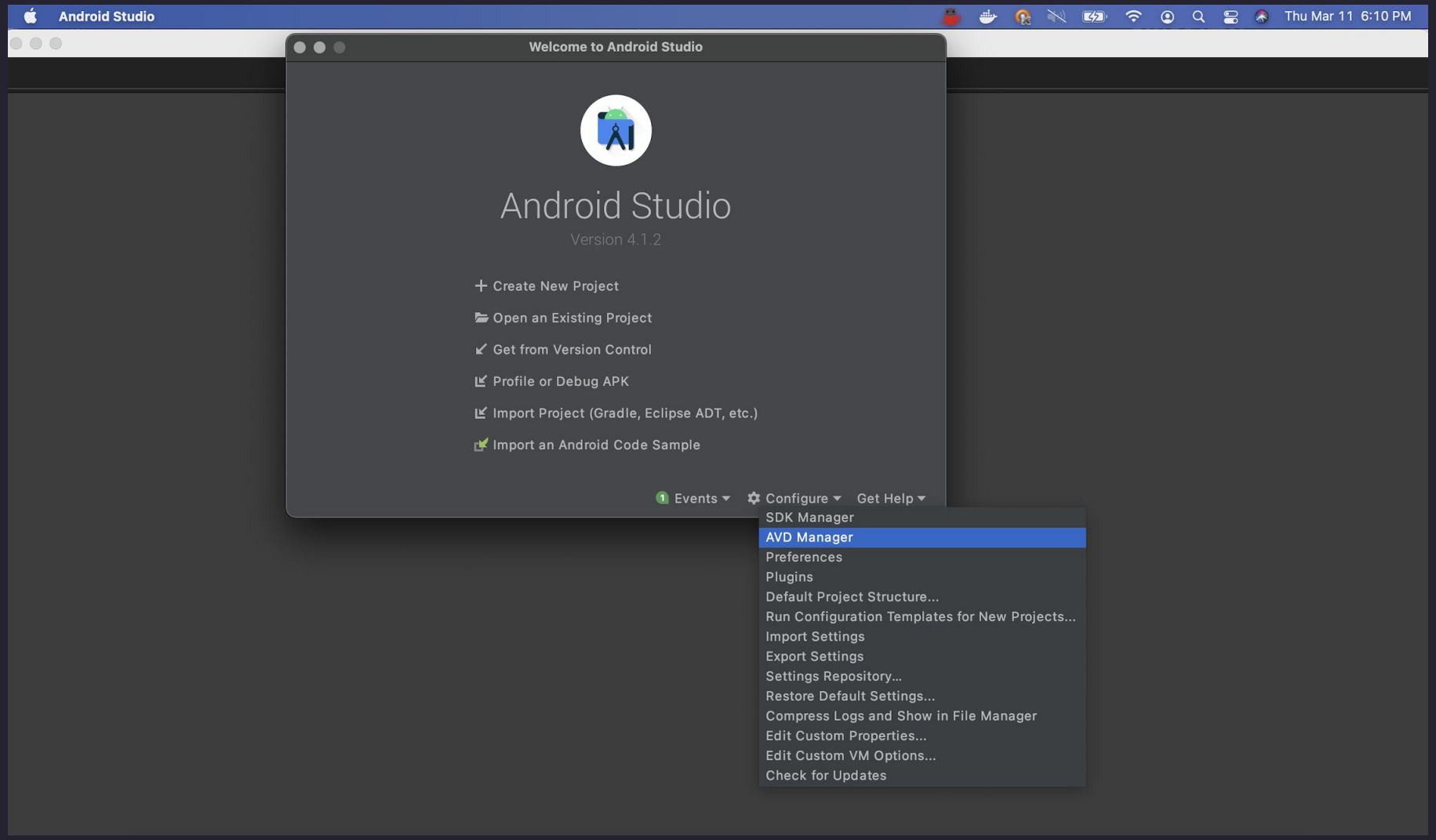
After following all of the instructions, all of the appium-doctor output should eventually show green check marks. This is a MacOS example.



Part One: Appium Installation Advice, Android Emulator, IOS Simulator Setups

Steps for setting up your Android Emulator.

- Start Android Studio.
- Click on the “Configure” button, and select AVD Manager.

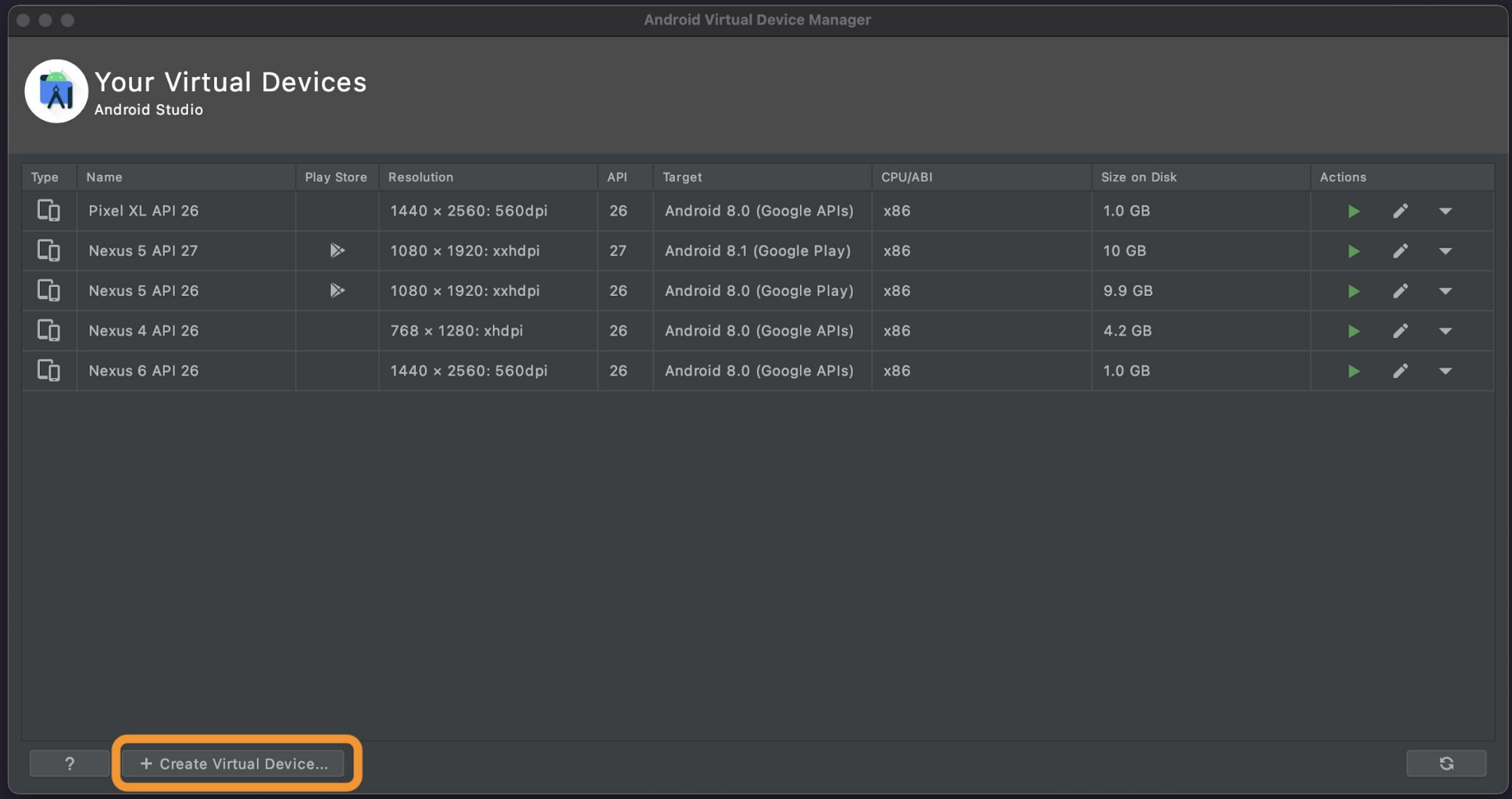


MAR 2021

Part One: Appium Installation Advice, Android Emulator, IOS Simulator Setups

Steps for setting up your Android Emulator.

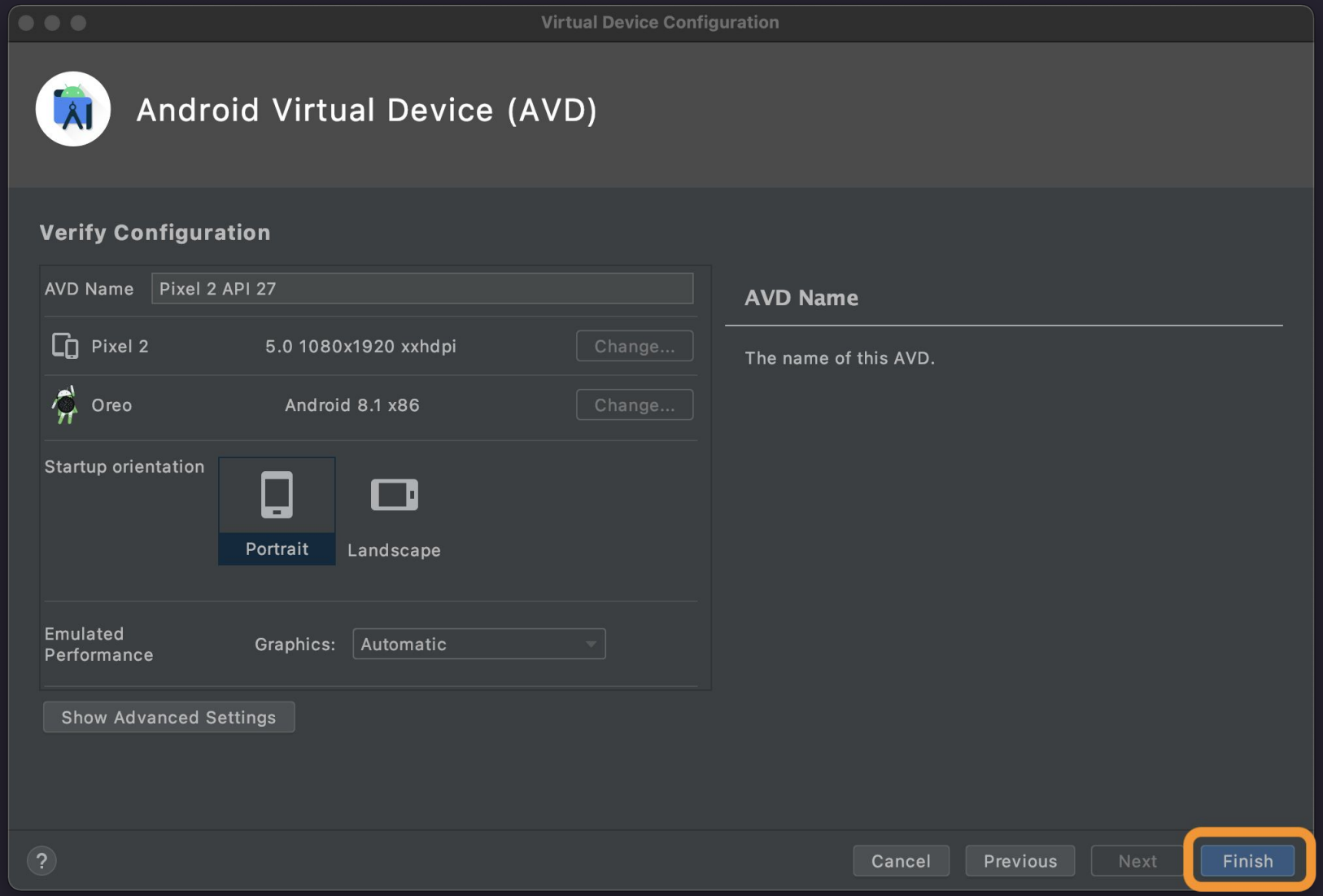
- From “AVD Manager”, click on the “Create Virtual Device” button.
- The “Virtual Device Configuration” will appear, select all of the options you need for your device.



Part One: Appium Installation Advice, Android Emulator, IOS Simulator Setups

Steps for setting up your Android Emulator.

- Click the “Finish” button from “Virtual Device Configuration”.
- Your new device should appear on the “AVD Manager” list.



MAR 2021

Part One: Appium Installation Advice, Android Emulator, IOS Simulator Setups

Steps for setting up your Android Emulator.

- Go to the root of the Getting-Started-Robotframework-AppiumLibrary-RoboCon-2021 repo.
- Open the start-specific-appium-example-workflows-for-workshop.sh file in an editor.
- Change the following (two places in the Bash script) to match the Android device that you have just set up...or...
- Workshop demonstrations are using a Nexus 5 API 27. You can also set up the same Android device.

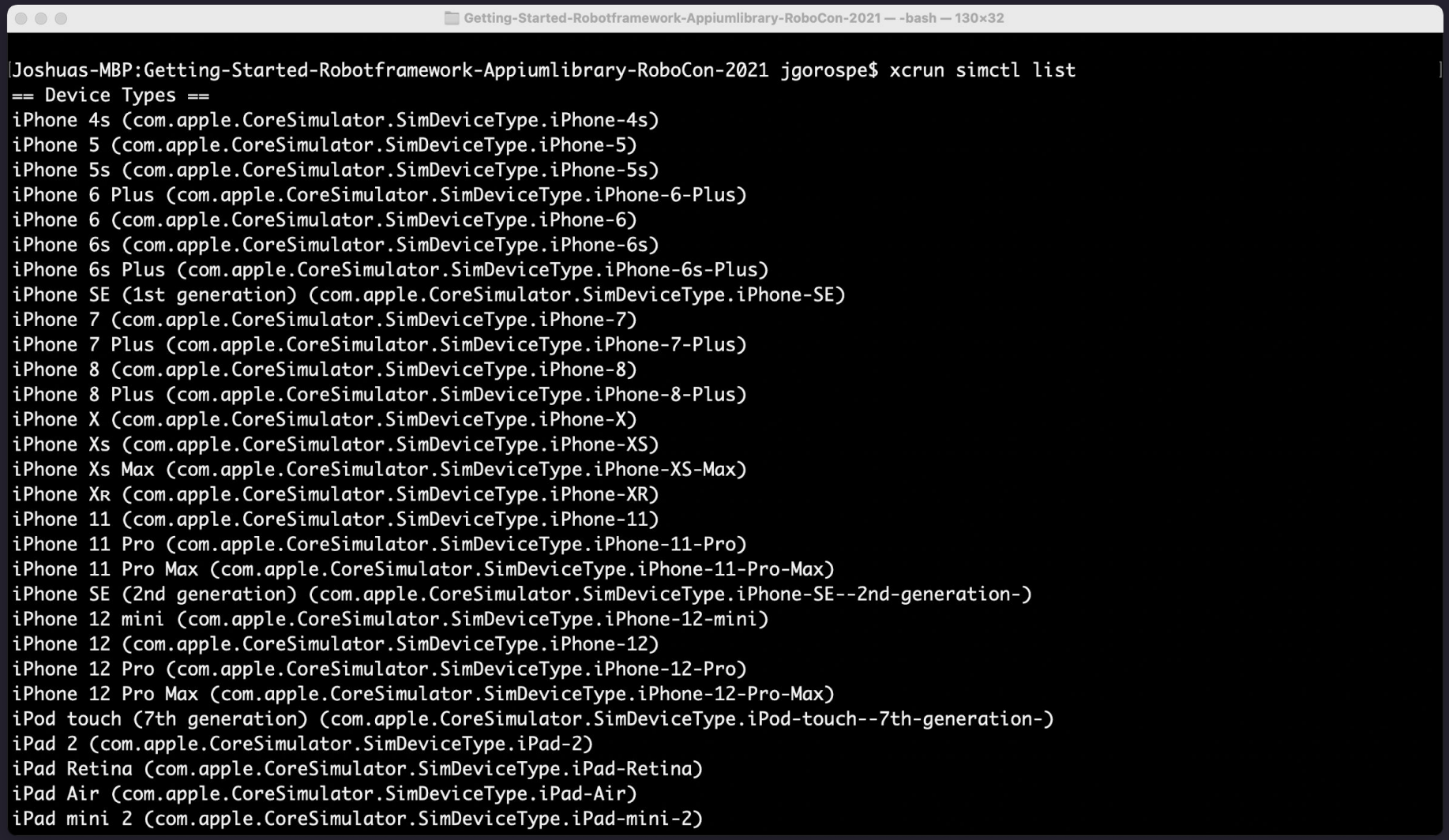
```
####
#### Copy and paste the following line for every new Android emulator device you want to add to every test setup. The following d
## "$HOME"/Library/Android/sdk/emulator/emulator -avd REPLACE_THIS_WITH_THE_NAME_OF_YOUR_ANDROID_EMULATOR -netdelay none -netspee
#### Copy and paste the following line for every new Android emulator device you want to add to every test setup. The following u
## "$HOME"/Library/Android/sdk/emulator/emulator -avd REPLACE_THIS_WITH_THE_NAME_OF_YOUR_ANDROID_EMULATOR -netdelay none -netspee
####
"$HOME"/Library/Android/sdk/emulator/emulator -avd Nexus_5_API_27 -netdelay none -netspeed full > $path/Workshop-Examples/Shared-
#"$HOME"/Library/Android/sdk/emulator/emulator -avd Nexus_6_API_26 -netdelay none -netspeed full > $path/Workshop-Examples/Shared
#"$HOME"/Library/Android/sdk/emulator/emulator -avd Pixel_XL_API_28 -netdelay none -netspeed full > $path/Workshop-Examples/Share
####
sleep 5s &&
appium -p 4723 --webdriveragent-port 8109 >> $path/Workshop-Examples/Shared-Resources/appium_output_log.txt 2>&1 & echo $! > $pat
appium -p 4724 -bp 5724 --allow-insecure chromedriver_autodownload --relaxed-security >> $path/Workshop-Examples/Shared-Resources
echo
echo
APPIUM_SERVER_PID1=$(cat ./Workshop-Examples/Shared-Resources/appium_server_PID1.txt)
```

MAR 2021

Part One: Appium Installation Advice, Android Emulator, IOS Simulator Setups

Steps for setting up your IOS Simulator.

- Installing XCode on your Mac machine will automatically install IOS Simulators for you.
- To check your available IOS Simulators, run this command -> xcrun simctl list
- Workshop demonstrations are using an iPad Pro (12.9-inch) (4th generation). Check it in your list.

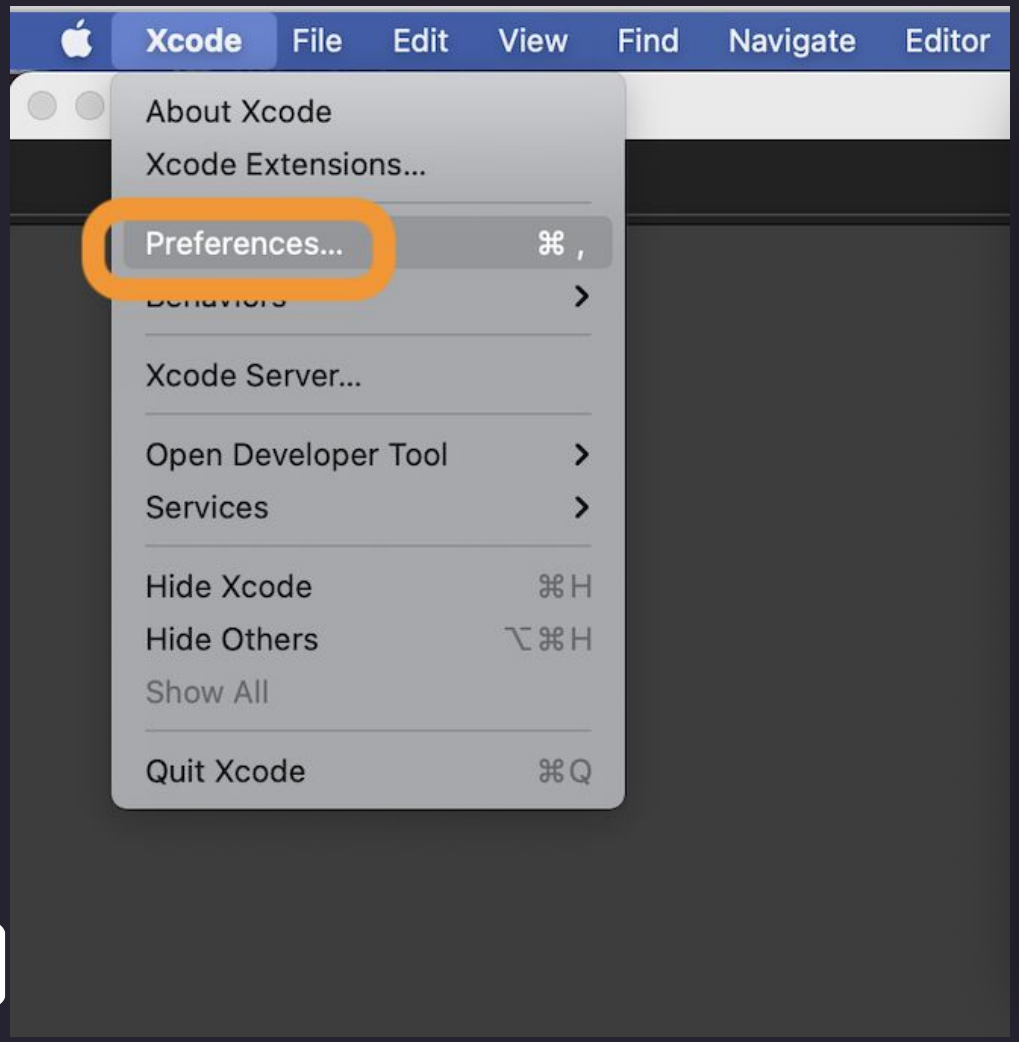


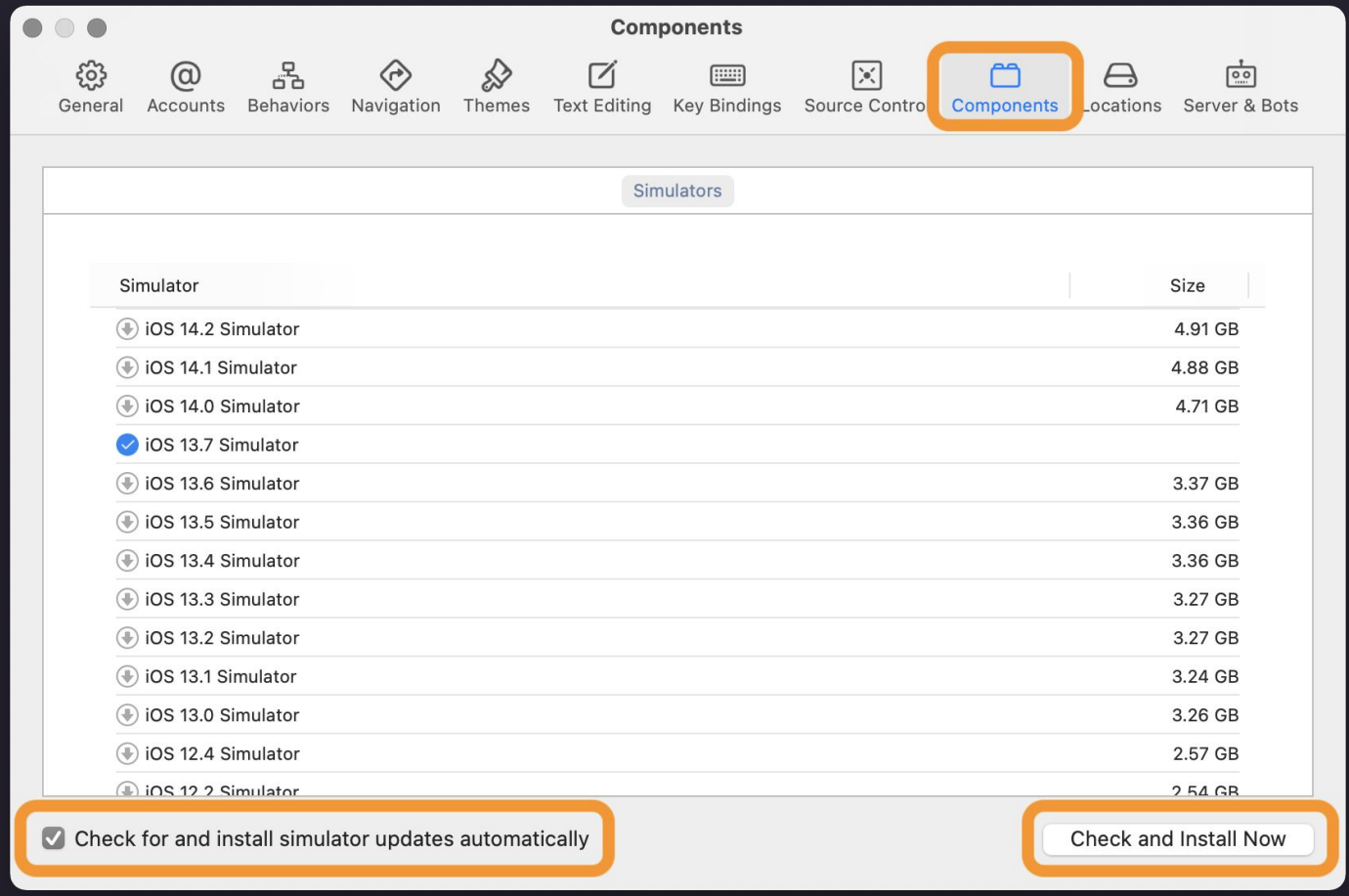
MAR 2021

Part One: Appium Installation Advice, Android Emulator, IOS Simulator Setups

Steps for setting up your IOS Simulator.

- XCode can also automatically check for new IOS updates by going to "Preferences", then "Components"
- You can enable the automatic update checkbox, or you can click "Check and Install Now.
- I installed iOS 13.7 and 14.4, which has iPad Pro (12.9-inch) (4th generation) in the xcrun simctl list output (previous slide).





Questions About Part One (installation, device setup, etc.)?

Part One: App automation workflows for IOS, and Android + adb shell

From your terminal or command-line.

- Go to the root of the Getting-Started-Robotframework-AppiumLibrary-RoboCon-2021 repo.
- Run the following commands first before running any of the examples for Part One.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh All-Appium-Tests-Teardown &&
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Appium-No-Proxy-Test-Setup

After the teardown and setup scripts successfully completed, you should see the Android device that was set up earlier appear on the screen.

- The following will install and run simple Wikipedia app tests on an iOS Simulator.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-IOS-App-Tests

Part One: App automation workflows for IOS, and Android + adb shell

The following will install and run simple Wikipedia app tests on an Android device.

- This example will also create a video recording.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_App_Recording_Screen_Capture

The following will run various adb shell commands and native Calculator app tests on an Android device. The commands below come from this comprehensive and useful GitHub list -> https://gist.github.com/Pulimet/5013acf2cd5b28e55036c82c91bd56d8

- This example will check the status of a specific Android app process that is active
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_PS_Command
- This example will check the file system contents for the current directory on the Android device.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_LS_Command
- This example will check that a specific package is installed on the Android device.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_Dumpsys_Grep_Command
- This example will check detailed information on all installed packages.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_Dumpsys_Package_Command
- This example will check a list of feature information about the Android device.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_PM_List_Command
- This example will check the current version of Android OS on the device.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_Build_Version_Command

Part One: App automation workflows for IOS, and Android + adb shell

The following will run various adb shell commands and native Calculator app tests on an Android device. The commands below come from this comprehensive and useful GitHub list -> https://gist.github.com/Pulimet/5013acf2cd5b28e55036c82c91bd56d8

- This example will check netstat (network statistics) for TCP connections going through the device.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_Netstat_Command
- This example will display the contacts list.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_Contacts_List_Command
- This example will to start and end a phone call on the device.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_Phone_Call_Command
- This example will prepare a text message from the device.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_Text_Message_Command
- This example will run logcat to collect the device's system logs in a file.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_Logcat_Command
- This example will run bugreport to collect the device's system logs, stack traces, and other helpful diagnostic data.
  - bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Android-Apps-And-Adb-Tests Android_Adb_Bugreport_Command

Part One: Browser automation for IOS and Android

The following will run Safari and Chrome browser automation examples.

- bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-IOS-And-Android-Mobile-Browsers-Test-Examples

Questions About Part One (examples etc.)?

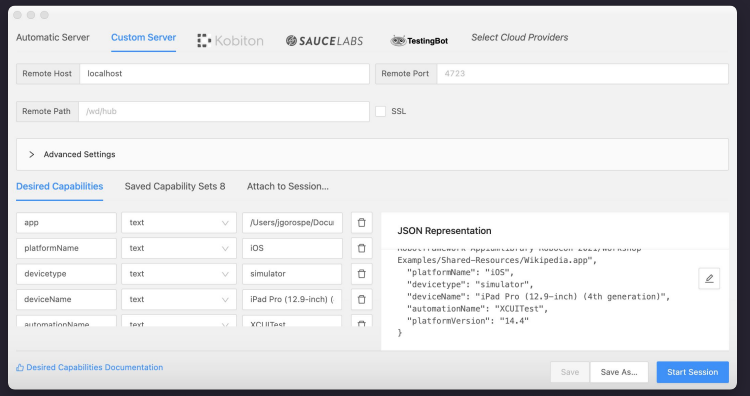
Part Two: Appium Desktop Walkthrough

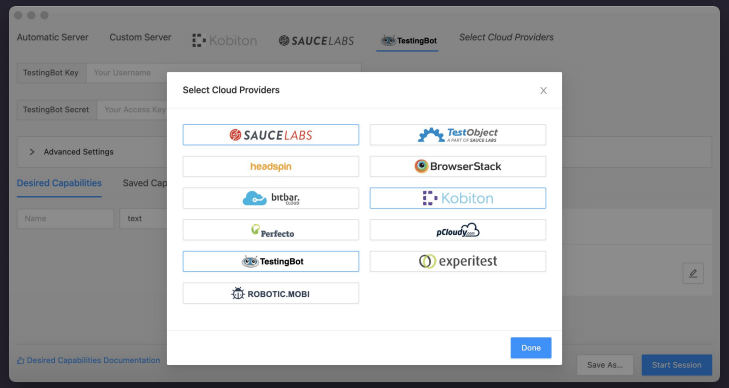
There are useful Appium Desktop features that can assist you while you are building out your automation.

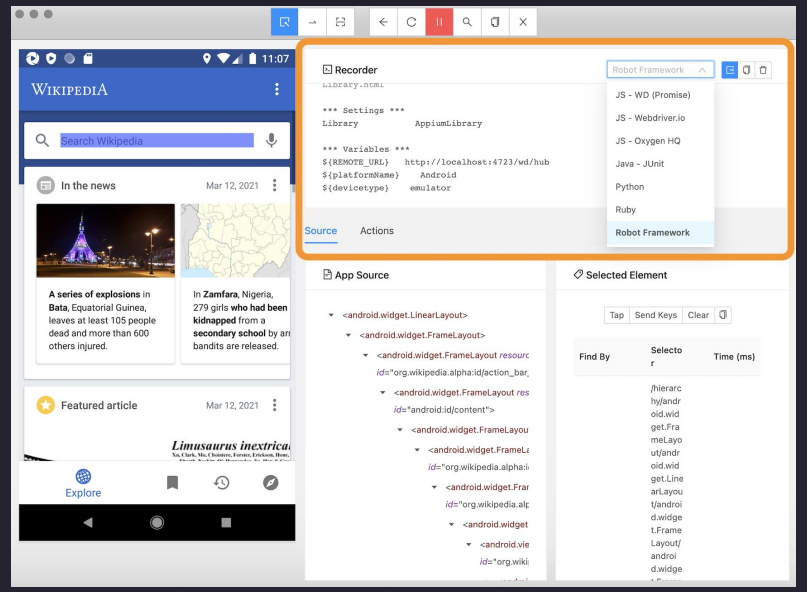
- Provides a convenient GUI for Appium Server and easy access to the following.
  - Capability sets
  - Connection options for mobile automation Cloud providers

- Appium Inspector -> gathers details about the UI elements of your app.

- Inspector Recording option generates [Appium automation code in Javascript, Python, Ruby, Java, and Robot Framework]







Part Two: Charles Proxy Combined With Robotframework-AppiumLibrary

Charles Proxy is a popular, well documented, and very useful web debugging proxy. It is not free software and requires a license key, but it's very affordable. It has a large list of builtin tools and features.

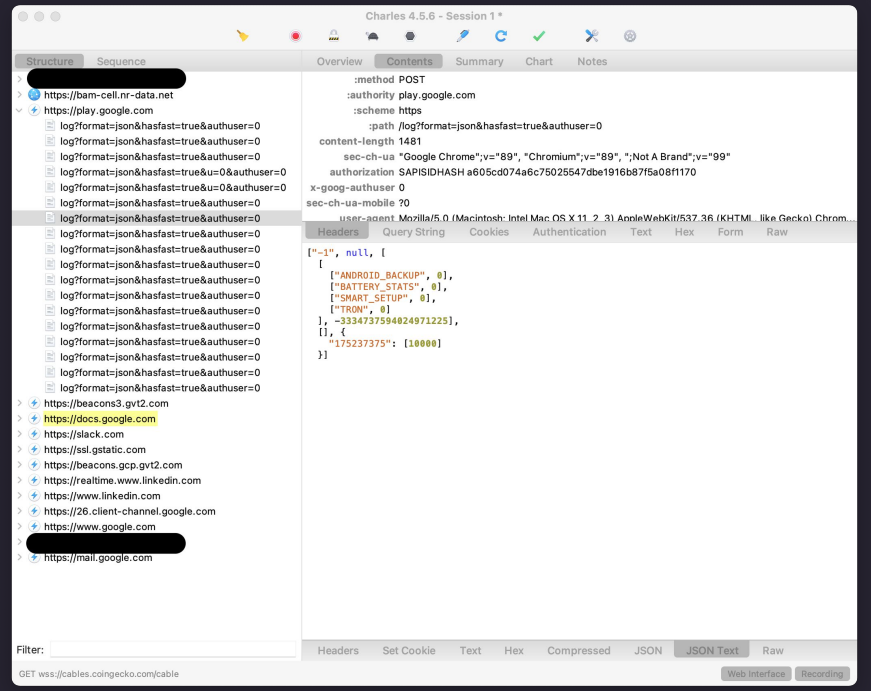
- https://www.charlesproxy.com/documentation/proxying/
- https://www.charlesproxy.com/documentation/tools/

The key features that we will be exploring in this Workshop are the following.

- Headless automation of Charles Proxy from its command-line tools.
- Capturing and recording all HTTP traffic going into any connected mobile device (real or Emulator/Simulator).

The following example commands requires a properly configured charles.config file, a generic file is provided in the Workshop repo. More information can be found here by searching for "charles.config" -> https://github.com/jg8481/Tool-Strategies-Lone-Testers-Test-Leadership-Congress-2019#ongoing-work

- bash ./start-specific-appium-example-workflows-for-workshop.sh All-Appium-Tests-Teardown &&
- bash ./start-specific-appium-example-workflows-for-workshop.sh Appium-Charles-Proxy-Test-Setup &&
- bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Charles-Proxy-IOS-And-Android-Mobile-Browsers-Test-Example



MAR 2021

Part Two: PaBot, Appium, And Graphwalker Combined With CPU Monitoring

The following examples will run an IOS and Android device in parallel using PaBot, a Robot Framework parallel test runner -> https://github.com/mkorpela/pabot

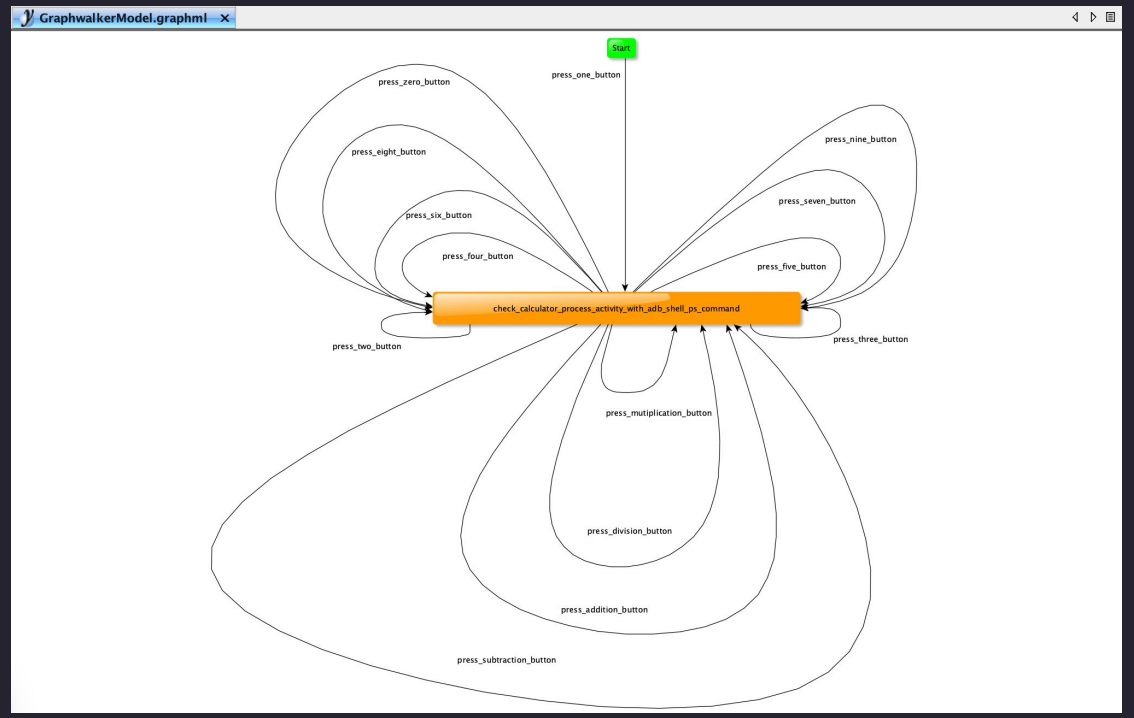
- On the IOS device, PaBot + Robotframework-AppiumLibrary will run the Wikipedia app in a loop.
- On the Android device PaBot + Robotframework-AppiumLibrary will run two types of tests in parallel.
  - Android Calculator app will be automated with a Model Based Testing approach that uses Graphwalker to generate different sequences of keywords based on a graph.
  - While the Android Graphwalker test is running, PaBot will monitor CPU usage using adb shell and Linux top command.

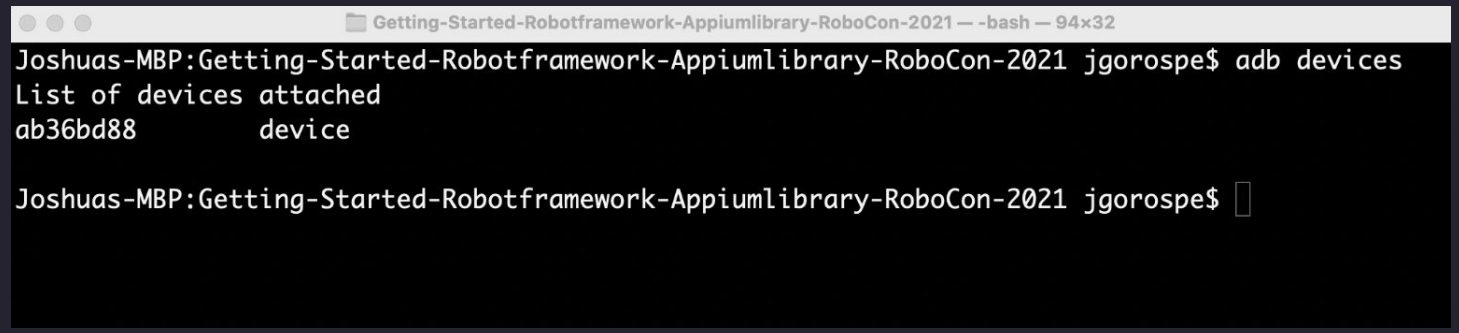
This command will run the PaBot automation.

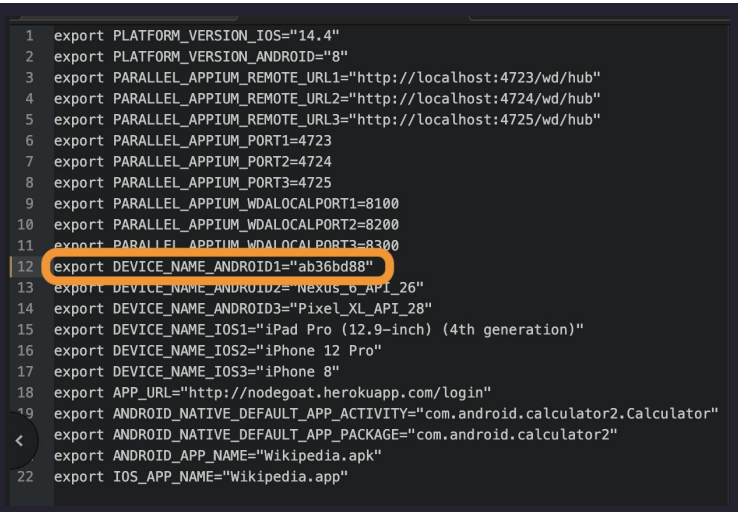
- `bash ./start-specific-appium-example-workflows-for-workshop.sh Robot-Framework-Parallel-IOS-Android-Tests`

Side Note: This same PaBot Android + Grahpwalker + adb shell CPU monitoring parallel running test can also be performed on a real Android device.

- Take your real Android device and enable the "Developer Options" by following the Android developers website instructions found here -> https://developer.android.com/studio/debug/dev-options
- Enable the "USB debugging" option in the Developer Options menu
- Plug in your device and run the "adb device" command from a terminal, then copy the alpha-numeric device name from adb into your DEVICE_NAME_ANDROID1" environment variable in your .env file







Questions About Part Two (examples etc.)?

Workshop Complete...

- Feel free to reach out to me on LinkedIn or Twitter with any high-level questions.
- I'm also open to presenting this live workshop again (not pre-recorded) through your Meetup Group.
- For deeper-dives into this content, I'm also open to perform live/pre-recorded small-scale online half-day workshops services for your team members or organization.

That's

ROBOCON 2021

#ROBOCON20

Robot Framework Foundation