Rx Mobile Automated Testing

Created by Chris Lupo, last modified by Christopher Helstern on May 13, 2020

Description:

Documentation for end to end automated regression tests of the Rx Mobile app features, which includes test bed setup and test execution.

Framework

Tests are written in Javascript using WebdriverIO. Refactor from the Java based Appium driver is ongoing, stay tuned.

Tests run using the Appium framework, which is a mobile flavor of Selenium. Using Appium with React-Native requires some workarounds to locate elements, but doesn't affect how the tests run. An Appium test run will start a web server that sends UIAutomator or XCUI commands to a mobile device running the app. This will either be local Android sim or an iOS sim)

Rx Test Structure

The tests have their own package, located at /uhc-react-native/packages/webdriver/

The webdriver directory has four subdirectories for rx test code:

1. Test directory: webdriver/rx/tests.
2. Page object models for the screen: webdriver/rx/screenobjects
3. common Rx UI interactions: webdriver/rx/interactions
4. webdriverIO config files: webdriver/configs

Prerequisites :

- Java 8+, NPM and Node (12 or latest stable version)
- Android and IOS local dev environment setup, follow UHC Mobile (LOWA) Developer Onboarding to complete the setup
- iOS (Xcode version needs to be 11.3.1)
  - Make sure the version Xcode is on version 11.3.1, it will not work with later versions (older versions can be found here: https://developer.apple.com/downloads/more/, will need to login with Apple ID credentials)
  - Current LOWA capability setups will require that the iOS device be set as a iPhone 11 - iOS 13.3, if this is not the default simulator with Xcode version 11.3.1, you can add the following line to open the right simulator:
    - Go to: packages/arcade/package.json
    - Set line 10: "i:e2e": "ENVFILE=.env.e2e react-native run-ios",
    - To: "i:e2e": "ENVFILE=.env.e2e react-native run-ios --simulator=\"iPhone 11 (13.3)\""`

Note: To verify your setup run below command in terminal:

- >Java -version
- >npm --version
- >node --version
  - --Make sure your node version is 12 or latest stable version:
  - Run: node --version ==> if less then 12
  - Run: sudo n lts

Appium Setup

To run tests locally, you will need to install Appium. Please make sure you have the latest version installed, as there are frequent bug fix updates.

1. `npm install -g appium (do not use yarn for this has to be npm)`
2. set `$JAVA_HOME` to your local flavor of Java (e.g. add this to `.bash_profile` -- `export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_191.jdk/Contents/Home`)
3. Add `$JAVA_HOME/bin` to your PATH (e.g. add this to .bash_profile --export PATH=$JAVA_HOME/bin:$PATH)
4. `brew install carthage`
5. (Optional)
   a. `nvm users: add to .bash_profile →`
      i. `export NODE_BINARY_PATH=$HOME/.nvm/versions/node/v8.15.0/bin/node` make sure to update this code to reference installed node version
      ii. `export APPIUM_BINARY_PATH="/usr/local/lib/node_modules/appium/build/lib/main.js"`
      iii. `https://github.com/appium/appium-dotnet-driver/wiki/How-to-start-an-AppiumDriver-locally`)
6. `install appium-doctor → `yarn global add appium-doctor` or via `npm install appium-doctor -g`// A tool to tell you what dependencies Appium is missing`
   a. run in terminal `appium-doctor` to display checklist of passing/failing dependencies (this will give you a diagnostic check that all necessary dependencies are set up properly)
7. See more at http://appium.io/docs/en/about-appium/getting-started/?lang=en
   a. Helpful point from this doc page is that you need to be using versions of node & Appium that were NOT installed using sudo, or that have the same permissions as your current user
   b. I was able to use the version of node that was installed through Homebrew, but I needed to change the permissions using `chown` from root to my user (i.e., `michael.young:admin`)

Running Tests

1. Clean and Start the metro bundler:
   a. cd ReactNativeMobileApp/uhc-react-native/
   b. Run: yarn clean
   c. Run: yarn install
   d. Run: yarn start
2. Build a debug version of the app without React warnings (which can block UI elements)
   a. cd ReactNativeMobileApp/uhc-react-native/
   b. Run: yarn e2e:i (iOS) or e2e:a (android)
3. `Run the Rx automation tests:`
   a. `cd ReactNativeMobileApp/uhc-react-native/packages/webdriver`

b. Run: yarn i --suite rx (iOS) or yarn a --suite rx (android)
    i. individual suites can be run as follows:
        1. pharmacy tests: yarn (a or i) --suite pharmacy
        2. drug search tests: yarn (a or i) --suite drug
        3. medicine cabinet tests: yarn (a or i) --suite medCab

References:

http://appium.io/docs/en/about-appium/getting-started/

https://github.com/appium/appium/blob/master/docs/en/writing-running-appium/caps.md

No labels