Migrating from Appium desktop app to Appium 1.6 CLI for testing iOS 10 with XCUITest

Test specialist Robert Gorter performs automated tests for an iOS application and has been struggling to support iOS 10. Since the iOS 10 release the UI Automation Instrument is deprecated and now it uses Xcode UI Testing.

To keep running his test scripts he had to make some major changes. In this blog post he shares the issues he ran into and describes how he overcame them.

I do test automation for an iOS application and have been struggling to support iOS 10. Since the iOS 10 release the UI Automation Instrument is deprecated and now it uses Xcode UI Testing. To keep running my test scripts I had to make some major changes. I'll save you the trouble of inventing the wheel for your own project by telling you some of my issues and how I overcame them.

What do you need?

You need Appium 1.6 CLI, WebDriverAgent and Element Inspector.

How to install Appium 1.6 CLI?

The installation of the CLI version of Appium is different from the desktop app. Before Appium version 1.6 I used the OSX app 1.4.13. Now I'm using the CLI version. My first question was, how do I install this? Using these commands: `$ npm install -g appium` `$ appium` After installing Appium, it's time to setup the environment starting with WebDriverAgent.

What's WebDriverAgent and how to setup the environment?

WebDriverAgent is a WebDriver server implementation for iOS that can be used to remote control iOS devices. It allows you to launch & kill applications, tap & scroll views or confirm view presence on a screen. This makes it a perfect tool for application end-to-end testing or general purpose device automation. It works by linking XCTest.framework and calling Apple's API to execute commands directly on a device. WebDriverAgent is developed and used at Facebook for end-to-end testing and is successfully adopted by Appium. https://github.com/facebook/WebDriverAgent

There are some dependencies necessary by the WebDriverAgent which can be found on Appium's Github page https://github.com/appium/appium-xcuitest-driver And the video below explains you how to setup iOS automation with XCUITest, I find this video tutorial extremely handy: https://www.youtube.com/watch?v=ySglJIrDVMQ&index=1&list=LLQmdNykQwiJSD_ayPykNYTA

A lot of commands are used in this video, I've listed them in the proper order below:

- `brew install ideviceinstaller`
- `brew install carthage`
- `npm install -g ios-deploy`
- `npm install -g deviceconsole`
- `gem install xcpretty`
- `brew install libimobiledevice --HEAD` (iOS 10)
- `mkdir -p Resources/WebDriverAgent.bundle`
- `sh ./Scripts/bootstrap.sh -d`
- `WebDriverAgent.xcodeproj`

The biggest obstacle that I overlooked for a long time after installing WebDriverAgent was the fact that I didn't sign the WebDriverAgent and WebDriverRunner with the same team and signing certificate as the App under test. This is mandatory. After setting everything up, it's time to start Appium.

How to start Appium?

The command I use to start Appium is:

```
appium --address 127.0.0.1 --port 4723 --default-capabilities '{"platformName":"iOS", "noReset":"true", "automationName":"XCUITest"}'
```

How can I inspect elements?

I found an incredible easy tool to inspect elements which is the Appium iOS Inspector made by Mykola-mokhnach and can be found here: https://github.com/mykola-mokhnach/Appium-iOS-Inspector You can debug your test script with breakpoints on the places you want to inspect and pick out the ID or Xpath with great ease.

Last but not least

After completing all these steps you should be able to write and run test scripts for iOS 10. If you still experience problems, leave a comment and I'll try to help you, also you can have a look at the forum, https://discuss.appium.io/