



JSConf Workshop

Adam Christian | @admc | Sauce Labs

appium is the cross-platform
solution for native and hybrid
mobile automation



appium introduction



Philosophy





R1. Test the same app you submit to the marketplace

R2. Write your tests in any language, using any framework

R3. Use a standard automation specification and API

R4. Build a large and thriving open-source community effort



Framework	R1	R2	R3	R4
calabash-ios				●
Frank				
UIAutomation				
ios-driver	●	●	●	
KeepItFunctional	●			
calabash-android				
MonkeyTalk				
Robotium			●	
UiAutomator				
selendroid		●	●	
appium				



Architecture

- Apple Instruments & UIAutomation for iOS
- Google UiAutomator for Android (4.2.1 up)
- Selendroid for older Android & hybrid
- Selenium WebDriver interface
- More platforms coming! (FirefoxOS, etc...)



Selenium WebDriver?

- the standard for browser automation, with libraries in every* language
- used every single day by thousands of developers familiar with its model
- W3C working draft



appium setup



Requirements (1/3)

- Xcode 4.6 with CLI tools and iOS 6.1
- Download Appium.app
- Node & NPM
 - `brew install node`



Requirements (2/3)

- Android Developer Tools \geq 21
 - <http://developer.android.com/sdk/index.html>
 - mv to /usr/local/adt
 - export ANDROID_HOME=/usr/local/adt/sdk
 - add (.bashrc, .zshrc, etc):
\$ANDROID_HOME/tools
\$ANDROID_HOME/platform-tools
to \$PATH



Requirements (3/3)

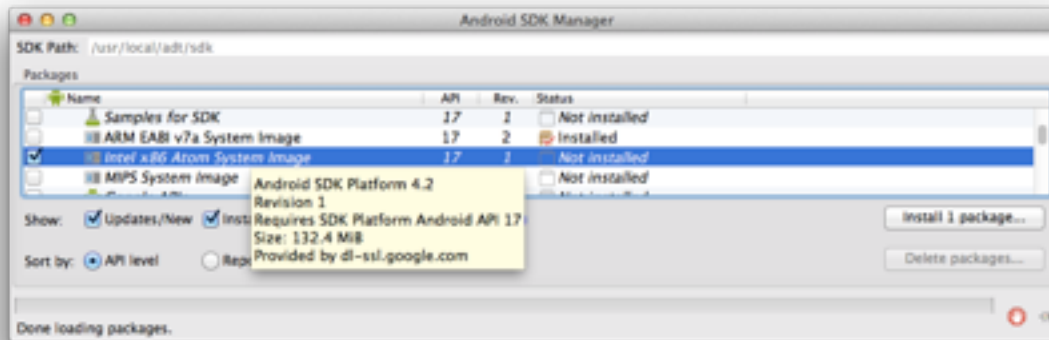
- Set up Sauce Labs demo account
- ```
export SAUCE_USERNAME="AppiumUser"
export SAUCE_PASSWORD="appiumrocks"
export SAUCE_ACCESS_KEY="e4c82f68-
cf00-40af-a8bc-5c78df0511a9"
```

\* This is for testing the Sauce Labs demo app -- not required for using Appium.



# Make an Android Device

- android
- Check 'Intel x86 Atom System Image' - Android (4.2.2)
- Click 'Install 1 package...'



- Tools > Manage AVDs
- New...



# Create the Image

- AVD Name: workshop
- Device: Nexus S
- Target: Android 4.2.2
- CPU: Intel/Atom
- Host GPU



# Install HAXM for Speed!

- open `/usr/local/adt/sdk/extras/intel/  
Hardware_Accelerated_Execution_Manager/  
IntelHAXM.dmg`



# Launch AVD

- In a new terminal window:
- `emulator @workshop -netfast`
- Go through the new device tour

```
emulator @workshop -netfast
HAX is working and emulator runs in fast virt mode
emulator: emulator window was out of view and was recentered
```

- `$ANDROID_HOME/sdk/tools/emulator @workshop -netfast (without env)`



# Get the workshop code

- `git clone https://github.com/appium/workshop.git appium-workshop`
- `cd appium-workshop`



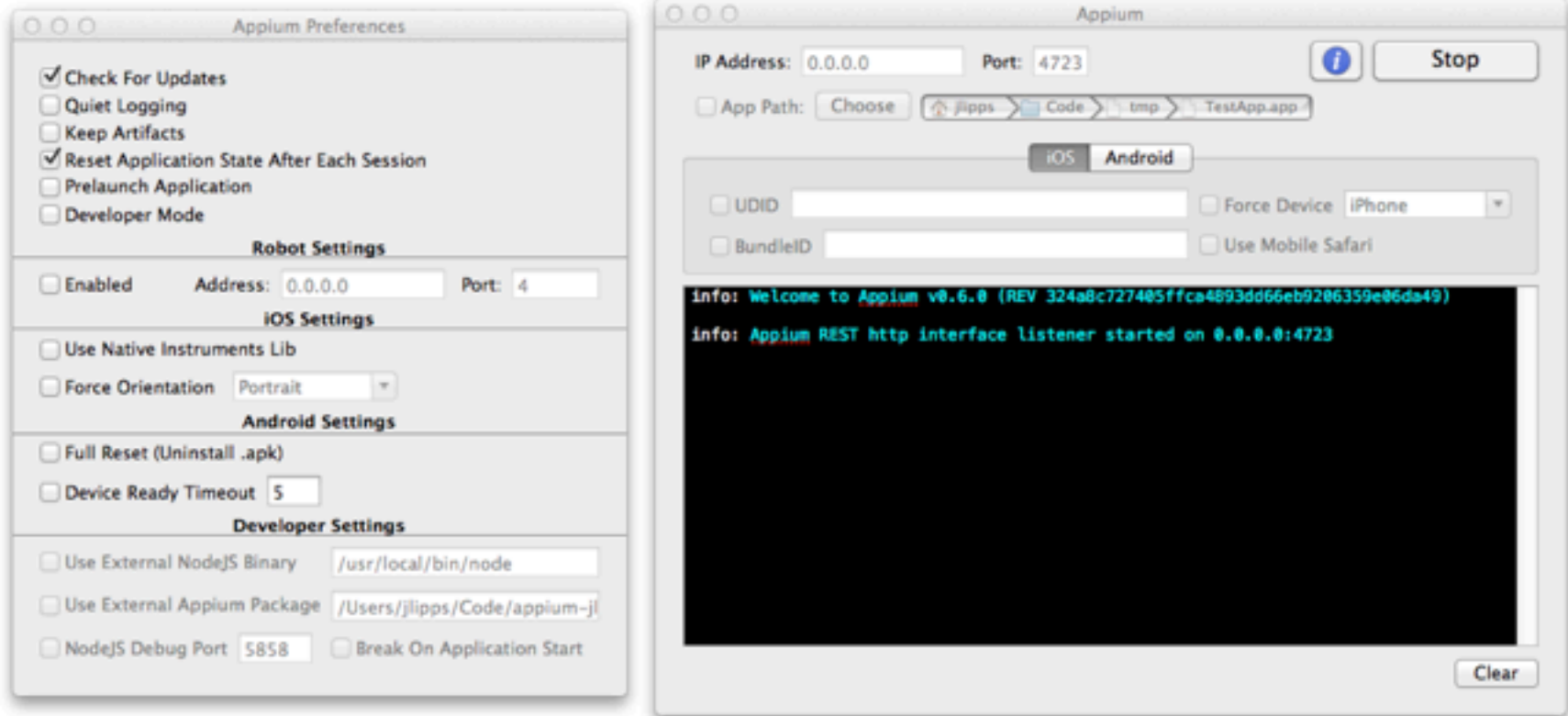


# Install dependencies

- `npm install -g mocha` # test runner
- `npm install .` # project deps
- `apps/download.sh` # download apps



# Launch Appium



- open /Applications/Appium.app



# Moment of truth...

- `mocha -t 60000 js/ios/step1.js`
- `mocha -t 60000 js/android/step1.js`

Also: step2 and step3



# appium test model



# Start a session

```
var wd = require('wd')
 , path = require('path')
 , relAppPath = "../apps/SauceDashboard.app.zip"
 , driver = wd.remote("localhost", 4723)
 , desiredCaps = {
 device: "iPhone Simulator"
 , app: path.resolve(__dirname, relAppPath)
 };

driver.init(desiredCaps, function(err, sessId) {
 // perform automation commands
});
```



# Find elements

```
driver.elementByName("Sign In", function(err, el) {
 // do something with 'Sign In' element
});
```

```
driver.elementByTagName("button", function(err, el) {
 // do something with button
});
```

```
driver.elementByXPath("//button[@text='Hi']", function(err, el) {
 // do something with button
});
```



# Interact with elements

```
driver.elementTagName("text", function(err, el) {
 el.text(function(err, text) {
 console.log("Text of element was " + text);
 el.click(function(err) {
 console.log("We clicked the element");
 });
 });
});
```



# Avoid rightward drift

```
driverSeries(driver, [
 function() { this.init(desiredCaps); },
 function() { this.elementByName("username"); },
 function() { this.res.sendKeys(USERNAME); },
 function() { this.elementByName("password"); },
 function() { this.res.sendKeys(PASSWORD); },
 function() { this.elementByName("login"); },
 function() { this.res.click(); },
 function() { this.quit(); }
], done);
```





# appium test building



# Pre-launch app to inspect

Appium Preferences

- Check For Updates
- Quiet Logging
- Keep Artifacts
- Reset Application State After Each Session
- Prelaunch Application
- Developer Mode

**Robot Settings**

Enabled    Address: 0.0.0.0    Port: 4

**iOS Settings**

Use Native Instruments Lib

Force Orientation    Portrait

**Android Settings**

Full Reset (Uninstall .apk)

Device Ready Timeout    5

**Developer Settings**

Use External NodeJS Binary    /usr/local/bin/node

Use External Appium Package    /Users/jlipps/Code/appium-jl

NodeJS Debug Port    5858     Break On Application Start

Appium

IP Address: 0.0.0.0    Port: 4723    Launch

App Path: Choose    jlipps > Code > appium-w > apps > SauceDashboard.app.zip

**iOS**    **Android**

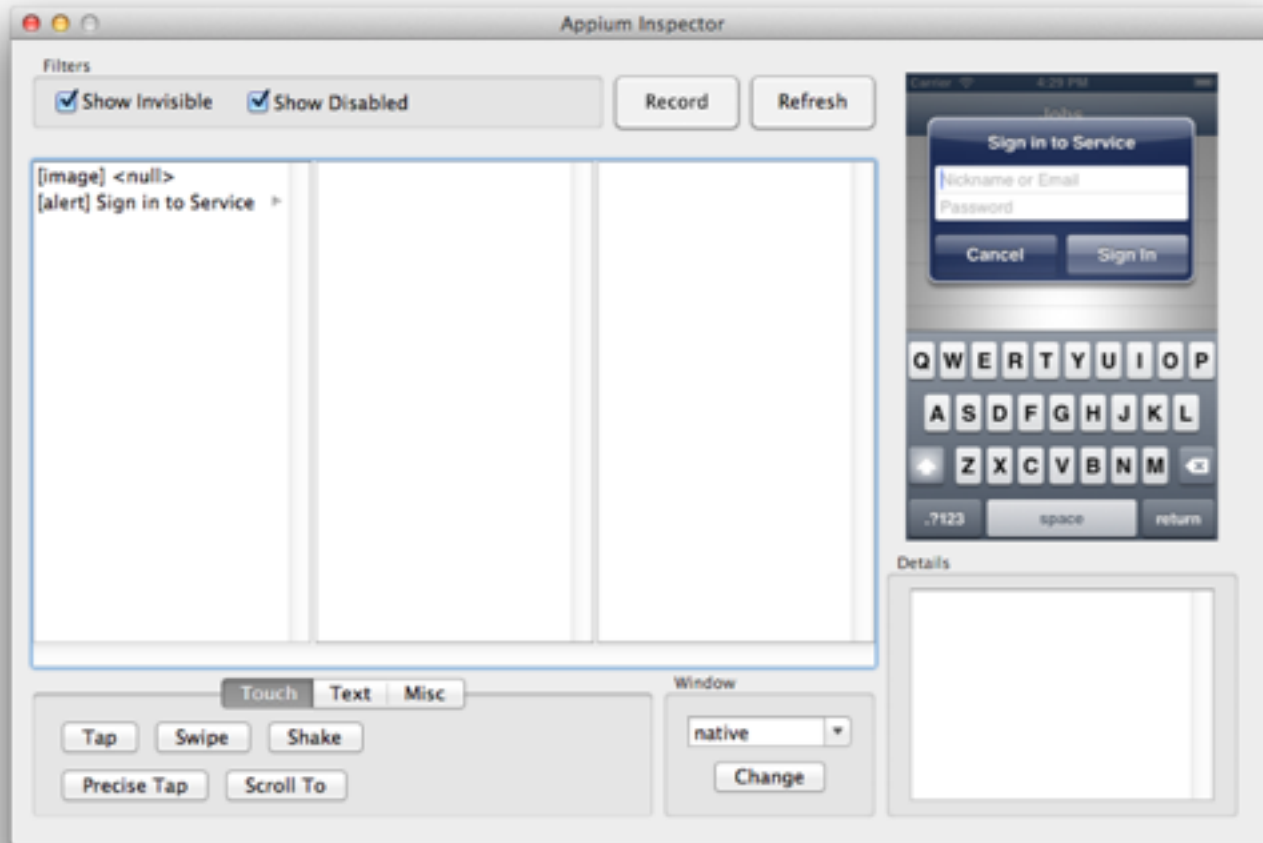
UDID     Force Device    iPhone

BundleID     Use Mobile Safari

Clear



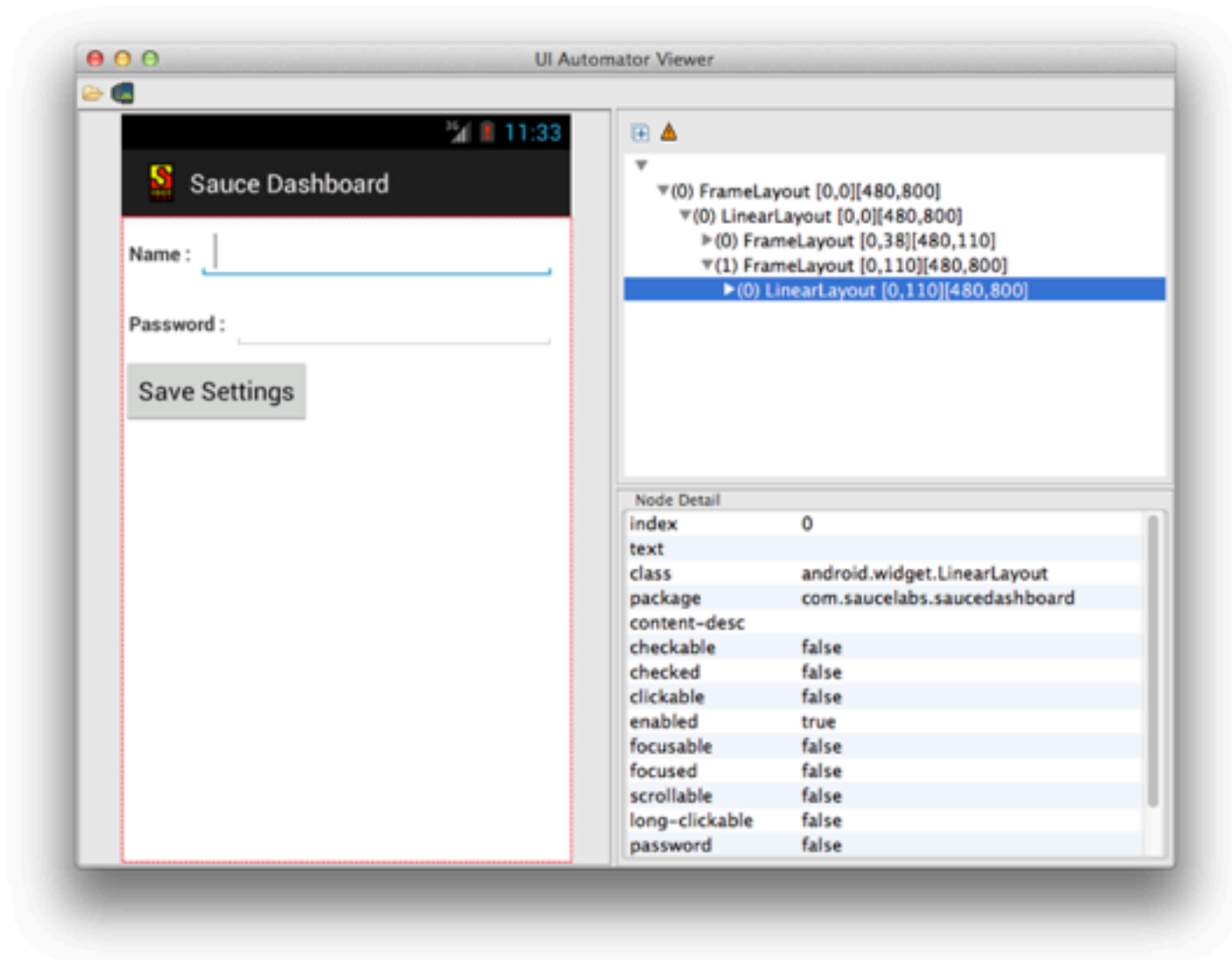
# Inspector helps!



**vim js/ios/step1.js**



# uiautomatorviewer for Android



**vim js/android/step1.js**



# appium real devices



\* Device must be in developer mode with usb debugging enabled.



# appium scale





**appium is great for local** test development, but has limitations when scaling up for use in CI



**Sauce Labs is great for scale**  
when you need to run a lot of  
**appium** tests in your build



 saucelabs.com/appium



# Run tests on Sauce

```
// LOCAL
var driver = wd.remote("localhost", 4723);
var relAppPath = "../apps/SauceDashboard.app.zip";
var desiredCaps = {
 device: "iPhone Simulator"
 , app: path.resolve(__dirname, relAppPath)
};

// SAUCE
var driver = wd.remote("http://ondemand.saucelabs.com", 80);
var appUrl = "http://appium.s3.amazonaws.com/SauceDashboard.app.zip";
var desiredCaps = {
 device: "iPhone Simulator"
 , app: appUrl
};
```



# Thanks!



appium.io

[github.com/appium/appium](https://github.com/appium/appium)

@AppiumDevs | @admc | @saucelabs