mobile automation made awesome

jQueryUK Workshop • Oxford, UK
May 16 2014

Jonathan Lipps • Director of Ecosystem & Integrations • Sauce Labs

@AppiumDevs • @jlipps • @saucelabs

Unit/Functional Mobile Testing with Appium and Sauce Labs

Ecosystem & Integrations

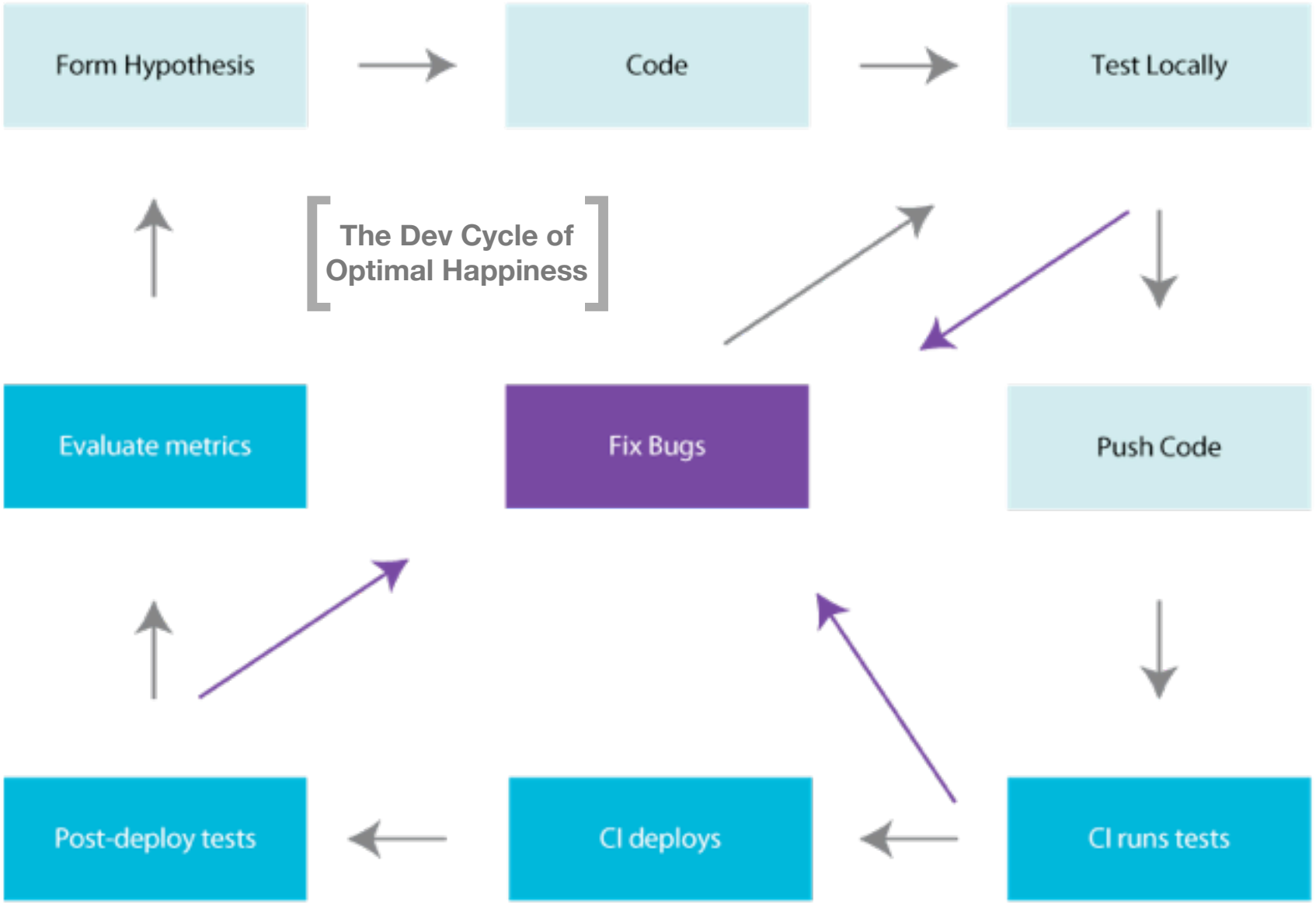
Project Lead & Architect

http://appium.io/jqueryuk.pdf

Jonathan Lipps • Director of Ecosystem & Integrations • Sauce Labs

@AppiumDevs • @jlipps • @saucelabs

appium introduction


Form Hypothesis
Code
Test Locally
The Dev Cycle of Optimal Happiness
Evaluate metrics
Fix Bugs
Push Code
Post-deploy tests
CI deploys
CI runs tests

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

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

runner.py (zsh)

appium.sh (bash) | runner.py (zsh)

```
~/C/demoize git:master ❯❯❯ mvimod
~/C/demoize git:master ❯❯❯ cd ~/Code/appium-demos/python; workon appium-dem
os
(appium-demos)~/C/a/python git:master ❯❯❯ android
(appium-demos)~/C/a/python git:master ❯❯❯ emulator @jlipps18demo -netfast
                                          ⊕
HAX is working and emulator runs in fast virt mode
(appium-demos)~/C/a/python git:master ❯❯❯ emulator @jlipps18demo -netfast -
scale 0.76                                ⊕
HAX is working and emulator runs in fast virt mode
```

```
1" 200 -
127.0.0.1 - - [04/Feb/2014 15:08:1
2] "GET /static/jquery-1.9.1-min.j
s HTTP/1.1" 200 -
127.0.0.1 - - [04/Feb/2014 15:08:1
2] "GET /static/demoize.js HTTP/1.
1" 200 -
127.0.0.1 - - [04/Feb/2014 15:08:1
2] "GET /favicon.ico HTTP/1.1" 404
 -
```

```
ET /static/github.css HTTP/1.1" 200 -
127.0.0.1 - - [04/Feb/2014 15:05:40] "G
ET /static/xyntax.css HTTP/1.1" 200 -
127.0.0.1 - - [04/Feb/2014 15:05:40] "G
ET /static/jquery-1.9.1-min.js HTTP/1.1
" 200 -
127.0.0.1 - - [04/Feb/2014 15:05:40] "G
ET /static/demoize.js HTTP/1.1" 200 -
127.0.0.1 - - [04/Feb/2014 15:05:40] "G
ET /favicon.ico HTTP/1.1" 404 -
```

```
~ ❯❯❯ cd ~/Code/appium-demos/python; workon appium-demos

(appium-demos)~/C/a/python git:master ❯❯❯ selenium-server
Feb 4, 2014 2:26:03 PM org.openqa.grid.selenium.GridLauncher main
INFO: Launching a standalone server
14:26:08.544 INFO - Java: Apple Inc. 20.8-b03-424
14:26:08.544 INFO - OS: Mac OS X 10.8.5 x86_64
14:26:08.564 INFO - v2.35.0, with Core v2.35.0. Built from revision c916b0d
14:26:08.738 INFO - RemoteWebDriver instances should connect to: http://127
.0.0.1:4444/wd/hub
14:26:08.739 INFO - Version Jetty/5.1.x
14:26:08.740 INFO - Started HttpContext[/selenium-server/driver,/selenium-s
erver/driver]
14:26:08.741 INFO - Started HttpContext[/selenium-server,/selenium-server]
14:26:08.741 INFO - Started HttpContext[/,/]
14:26:08.769 INFO - Started org.openqa.jetty.jetty.servlet.ServletHandler@1
5aaf8b3
14:26:08.769 INFO - Started HttpContext[/wd,/wd]
14:26:08.778 INFO - Started SocketListener on 0.0.0.0:4444
14:26:08.778 INFO - Started org.openqa.jetty.jetty.Server@acfec48
```

```
(appium-demos)~/C/a/python git:master ❯❯❯ ./runner.py --android && ./runner
.py --ios
Starting demo
```

appium architecture

Automation Orchestra

Apple **Instruments** & **UIAutomation** for iOS

Google **UiAutomator** for Android (4.2.1 up)

Selendroid for older Android

WebDriver interface

appium is an HTTP server that creates and handles **WebDriver sessions**

appium extends the WebDriver protocol with **mobile-specific** behaviors

appium setup

Requirements (1/2)

- Mac (10.8/10.9)
  - Android automation works on PC/Linux too
- Node >= 0.10
- Xcode 5.1 with CLI tools and iOS 7.1

Requirements (2/2)

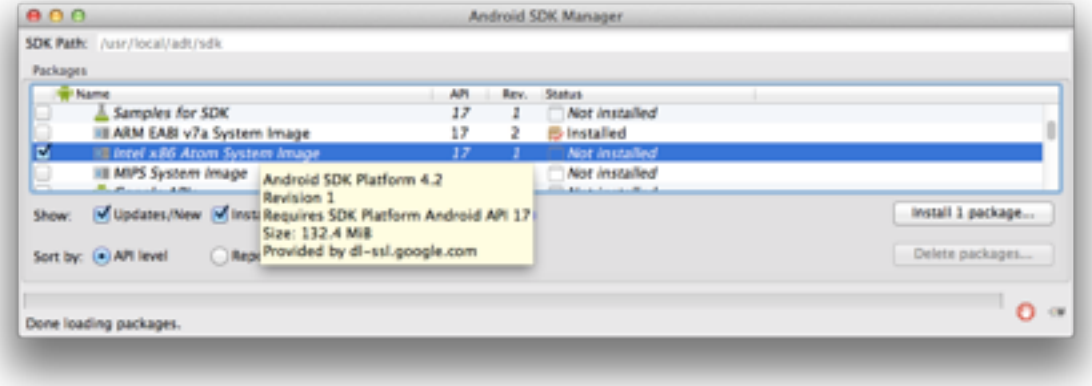
- Android Developer Tools >= 22
  - http://developer.android.com/sdk/index.html
  - mv to `/usr/local/adt`
  - `export ANDROID_HOME=/usr/local/adt/sdk`
  - add (.bashrc, .zshrc, etc):
    `export PATH="$PATH:$ANDROID_HOME/tools:`
    `$ANDROID_HOME/platform-tools"`

Install HAXM for Android Speed!

- `open /usr/local/adt/sdk/extras/intel/Hardware_Accelerated_Execution_Manager/IntelHAXM.dmg`
- https://software.intel.com/en-us/android/articles/intel-hardware-accelerated-execution-manager

Make an Android Device

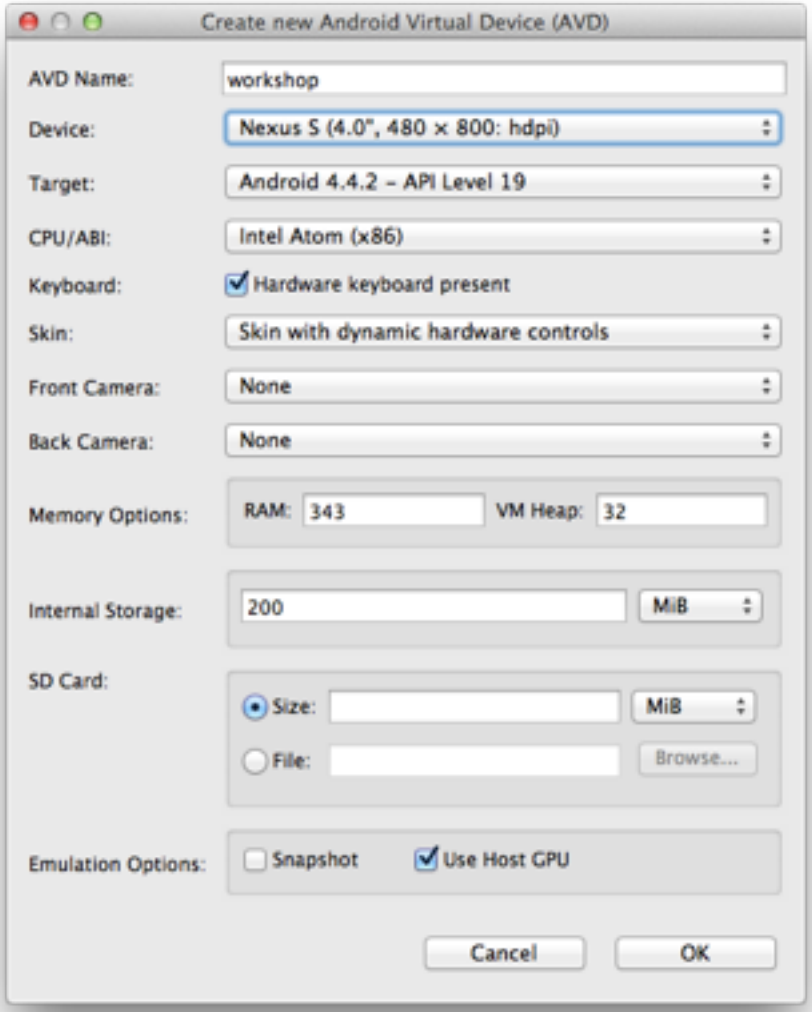
- `android`
- Check ‘Intel x86 Atom System Image’ - Android (4.4)
- Click ‘Install 1 package...’



- Tools > Manage AVDs
- New...

Create the Image

- AVD Name: workshop
- Device: Nexus S
- Target: Android 4.4
- CPU: Intel/Atom
- Skin: hw controls
- Host GPU



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/jlipps/jqueryuk-workshop-2014.git`
- `cd jqueryuk-workshop-2014`

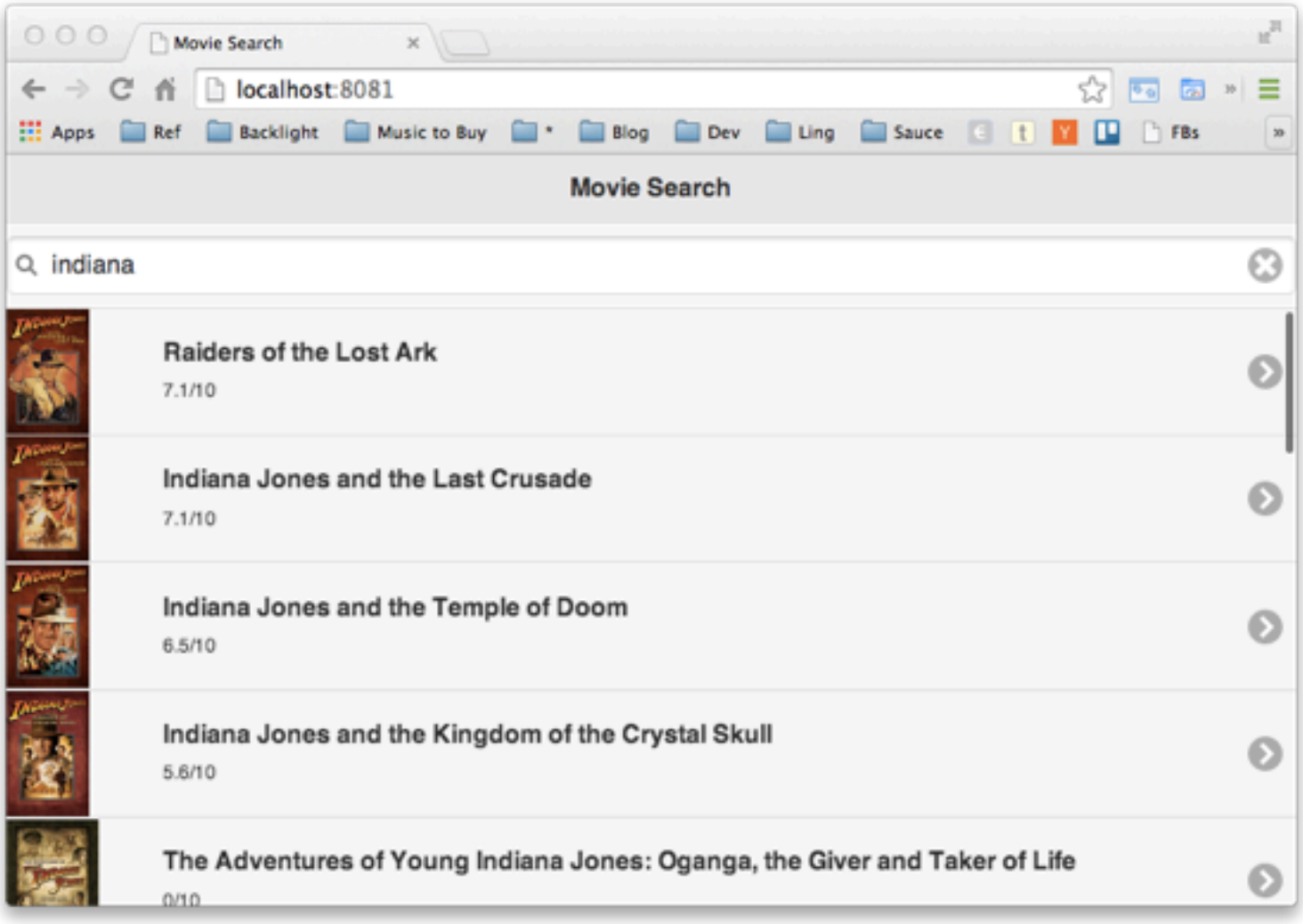
Install dependencies

- `npm install -g appium  # no sudo!`
- `npm install -g cordova`
- `npm install -g mocha`
- `npm install .`

unit tests

Run local server

- `node server.js`
- `# visit http://localhost:8081`


Movie Search
localhost:8081
Apps
Ref
Backlight
Music to Buy
Blog
Dev
Ling
Sauce
FBs
Movie Search
indiana
Raiders of the Lost Ark
7.1/10
Indiana Jones and the Last Crusade
7.1/10
Indiana Jones and the Temple of Doom
6.5/10
Indiana Jones and the Kingdom of the Crystal Skull
5.6/10
The Adventures of Young Indiana Jones: Oganga, the Giver and Taker of Life
0/10

Run QUnit tests

- http://localhost:8081/test.html
- # moviesearch/www/test.html
- # moviesearch/www/test.js

Set up Sauce Labs env vars

- http://saucelabs.com/signup/plan/free
- http://saucelabs.com/account
- # add to .bashrc or equivalent
- export SAUCE_USERNAME=”myusername”
- export SAUCE_ACCESS_KEY=”xxxxxxxx”

Get Sauce Connect

- https://saucelabs.com/docs/connect
- cp ~/Downloads/sc-4.2-osx/bin/sc \
  /usr/local/bin

Start Sauce Connect

- `sc -u $SAUCE_USERNAME -k $SAUCE_ACCESS_KEY`

Run JS unit tests on Sauce Labs

- `./test/jsunit.sh`
- http://saucelabs.com/tests

appium test model

Start/stop a session

```
var wd = require('wd');

describe('MovieSearch app', function () {
  var driver;

  before(function (done) {
    driver = wd.promiseChainRemote('localhost', 4723);
    driver.init({
      platformName: 'iOS',
      platformVersion: '7.1',
      deviceName: 'iPhone Simulator',
      app: '/path/to/my.app'
    }).nodeify(done);
  });

  after(function (done) {
    driver.quit().nodeify(done);
  });
});
```

Find & Interact with Elements

```
driver
  .elementByClassName("UIAButton")
  .text() // get its text
  .then(function (text) { console.log(text); })
  .click() // click element

driver
  .elementByAccessibilityId("username")
  .sendKeys("jlipps") // type text into a field
```

Automate a WebView

```
driver
   .contexts() // ['NATIVE_APP', 'WEBVIEW_1']
   .context("WEBVIEW_1")
   .elementByCss("a.clickme")
   .click()
```

appium tests

Build & run sample apps

- `./go_ios.sh`
- `./go_android.sh`

Launch Appium

- `sudo authorize_ios`
- `appium`

Moment of truth...

- `mocha -t 90000 -R spec test/ios.js`
- `mocha -t 90000 -R spec test/android.js`

Upload app to Sauce Storage

- `./test/upload.sh`
- `# or use pre-uploaded app url`

Run Appium tests on Sauce

- `SAUCE=1 mocha -t 90000 -R spec test/ios.js`

Questions?

http://appium.io
https://github.com/appium/appium
@AppiumDevs • @jlipps • @saucelabs

Thanks!

http://appium.io
https://github.com/appium/appium
@AppiumDevs • @jlipps • @saucelabs