



Chinese Mobile Testing Summit • Beijing, CN
11 July 2015

Jonathan Lipps • Director of Ecosystem & Integrations • Sauce Labs

@AppiumDevs • @jlipps • @saucelabs

Appium: Open Source Mobile Automation



Director of Engineering,
Ecosystems



Project Lead &
Architect

Jonathan Lipps • Director of Ecosystem & Integrations • Sauce Labs

@AppiumDevs • @jlipps • @saucelabs

appium introduction

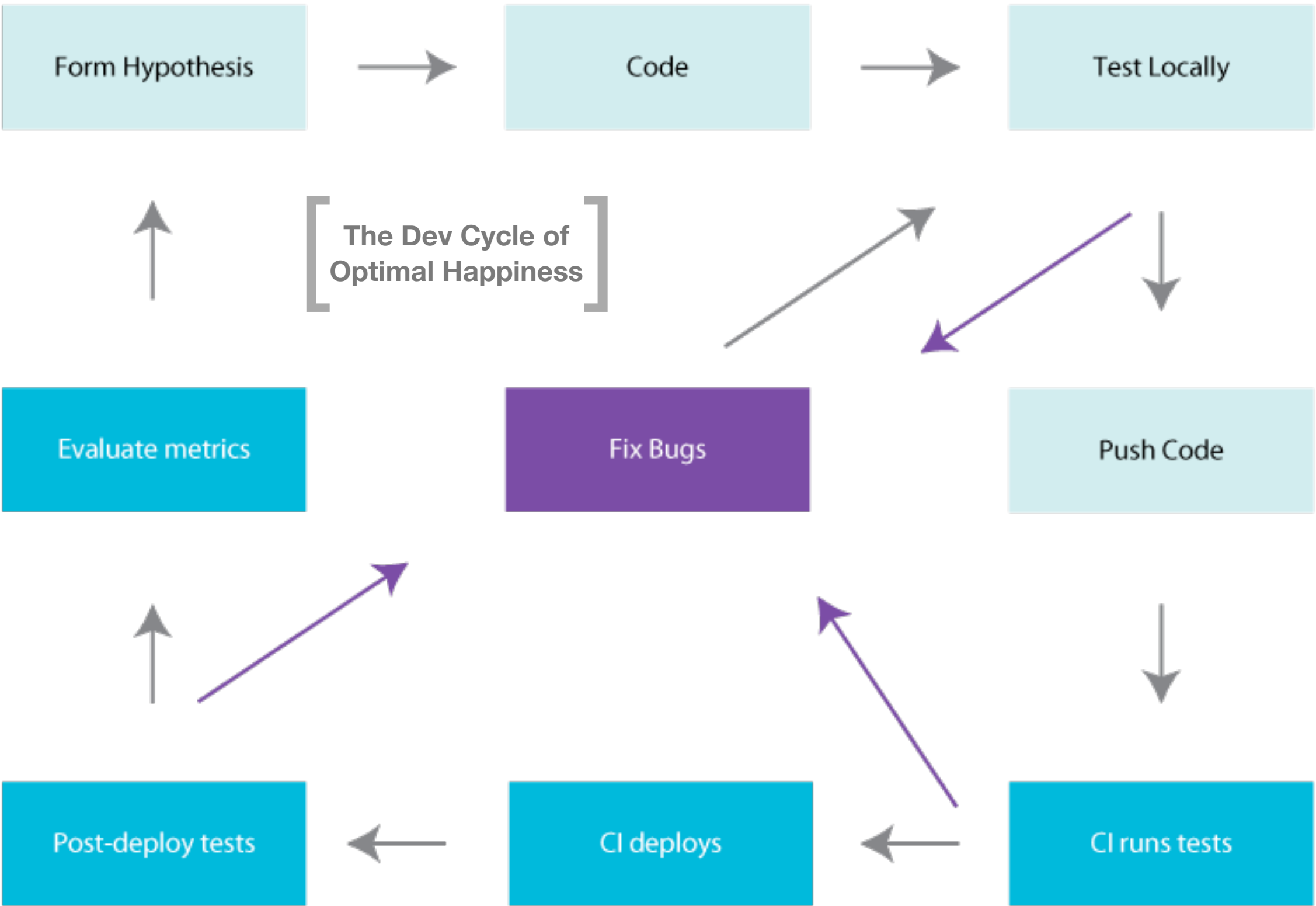


Mobile is here to stay. So how
do we **scale mobile quality**?



Automated testing is the solution
for a fast dev cycle that maintains
high quality





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



appium raison d'être



The appium 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



Platform Support

Real devices  

Simulators  

Native apps  

Hybrid apps  

Mobile web  



Over **65** releases

Over **5,600** commits

3,000 issues closed

1,900 pull requests merged

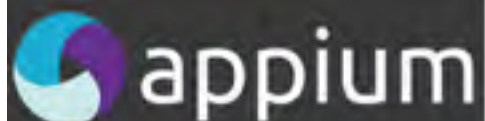
2,400 stars

1,600 forks

Over **160** contributors

OSS Rookie of the Year, **Bossie** award





About Appium

About Appium

Appium 介绍

Setting up Appium

Intel® 硬件加速器管理

部署iOS app 到手机上

Appium支持的平台

Appium在真机上

在 Linux 上运行 Appium

在 Mac OS X 上使用 Appium

在Windows上运行Appium

Running Appium tests

执行测试

Appium 服务器参数

Advanced Concepts

Selenium Grid

自动化混合应用

把appium 0.18.x上的测试用例...

Ruby bindings

Python bindings

Java bindings

JavaScript bindings

PHP bindings

C# bindings

Appium home page

客户端类库列表及Appium服务端支持

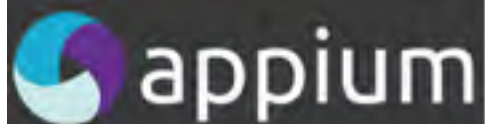
这些类库封装了标准Selenium客户端类库，为用户提供所有常见的JSON 格式selenium命令以及额外的移动设备控制相关的命令，如多点触控手势和屏幕朝向。

Appium客户端类库实现了Mobile JSON Wire Protocol（一个标准协议的官方扩展草稿）和W3C Webdriver spec（一个传输不可预知的自动化协议，该协议定义了MultiAction 接口）的元素。

Appium 服务端定义了官方协议的扩展，为Appium 用户提供了方便的接口来执行各种设备动作，例如在测试过程中安装/卸载app。这就是为什么我们需要Appium 特定的客户端，而不是通用的Selenium 客户端。当然，Appium 客户端类库只是增加了一些功能，而实际上这些功能就是简单的扩展了Selenium 客户端，所以他们仍然可以用来运行通用的selenium会话。

语言/框架	Github版本库以及安装指南
Ruby	https://github.com/appium/ruby_lib
Python	https://github.com/appium/python-client
Java	https://github.com/appium/java-client
JavaScript (Node.js)	https://github.com/admc/wd
Objective C	https://github.com/appium/selenium-objective-c
PHP	https://github.com/appium/php-client
C# (.NET)	https://github.com/appium/appium-dotnet-driver
RobotFramework	https://github.com/jollychang/robotframework-appiumlibrary





About Appium

About Appium

Appium 介绍

Setting up Appium

Intel® 硬件加速器管理

部署iOS app 到手机上

Appium支持的平台

Appium在真机上

在 Linux 上运行 Appium

在 Mac OS X 上使用 Appium

在Windows上运行Appium

Running Appium tests

执行测试

Appium 服务器参数

Advanced Concepts

Selenium Grid

自动化混合应用

把appium 0.18.x上的测试用例...

Ruby bindings

Python bindings

Java bindings

JavaScript bindings

PHP bindings

C# bindings

Appium home page

客户端类库列表及Appium服务端支持

这些类库封装了标准Selenium客户端类库，为用户提供所有常见的JSON 格式selenium命令以及额外的移动设备控制相关的命令，如多点触控手势和屏幕朝向。

Appium客户端类库实现了Mobile JSON Wire Protocol（一个标准协议的官方扩展草稿）和W3C Webdriver spec（一个传输不可预知的自动化协议，该协议定义了MultiAction 接口）的元素。

Appium 服务端定义了官方协议的扩展，为Appium 用户提供了方便的接口来执行各种设备动作，例如在测试过程中安装/卸载app。这就是为什么我们需要Appium 特定的客户端，而不是通用的Selenium 客户端。当然，Appium 客户端类库扩展了Selenium 客户端，所以他们仍然可以用来运行通用的selenium命令。

语言

Ruby [selenium-ruby](#)

Python [selenium-python-client](#)

Java <https://github.com/appium/java-client>

JavaScript (Node.js) <https://github.com/admc/wd>

Objective C <https://github.com/appium/selenium-objective-c>

PHP <https://github.com/appium/php-client>

C# (.NET) <https://github.com/appium/appium-dotnet-driver>

RobotFramework <https://github.com/jollychang/robotframework-appiumlibrary>

谢谢



Demos





```
runner.py (zsh)
appium.sh (bash) runner.py (zsh)
~/Code/appium-demos/python git:master >>> mvimod
~/Code/appium-demos/python git:master >>> cd ~/Code/appium-demos/python; workon appium-demos
(appium-demos)~/Code/appium-demos/python git:master >>> android
(appium-demos)~/Code/appium-demos/python git:master >>> emulator @jlipps18demo -netfast
HAX is working and emulator runs in fast virt mode
(appium-demos)~/Code/appium-demos/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 15:08:12] "GET /static/jquery-1.9.1-min.js HTTP/1.1" 200 -
127.0.0.1 - - [04/Feb/2014 15:08:12] "GET /static/demoize.js HTTP/1.1" 200 -
127.0.0.1 - - [04/Feb/2014 15:08:12] "GET /static/demoize.js HTTP/1.1" 200 -
127.0.0.1 - - [04/Feb/2014 15:08:12] "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] "GET /static/syntax.css HTTP/1.1" 200 -
127.0.0.1 - - [04/Feb/2014 15:05:40] "GET /static/jquery-1.9.1-min.js HTTP/1.1" 200 -
127.0.0.1 - - [04/Feb/2014 15:05:40] "GET /static/demoize.js HTTP/1.1" 200 -
127.0.0.1 - - [04/Feb/2014 15:05:40] "GET /favicon.ico HTTP/1.1" 404 -

~ >>> cd ~/Code/appium-demos/python; workon appium-demos
(appium-demos)~/Code/appium-demos/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 c916b9d
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-server/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@15aaf0b3
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)~/Code/appium-demos/python git:master >>> ./runner.py --android && ./runner.py --ios
Starting demo
```


...xamples/node

~/C/a/s/e/node git:master >>> |

...ppium-jlipps

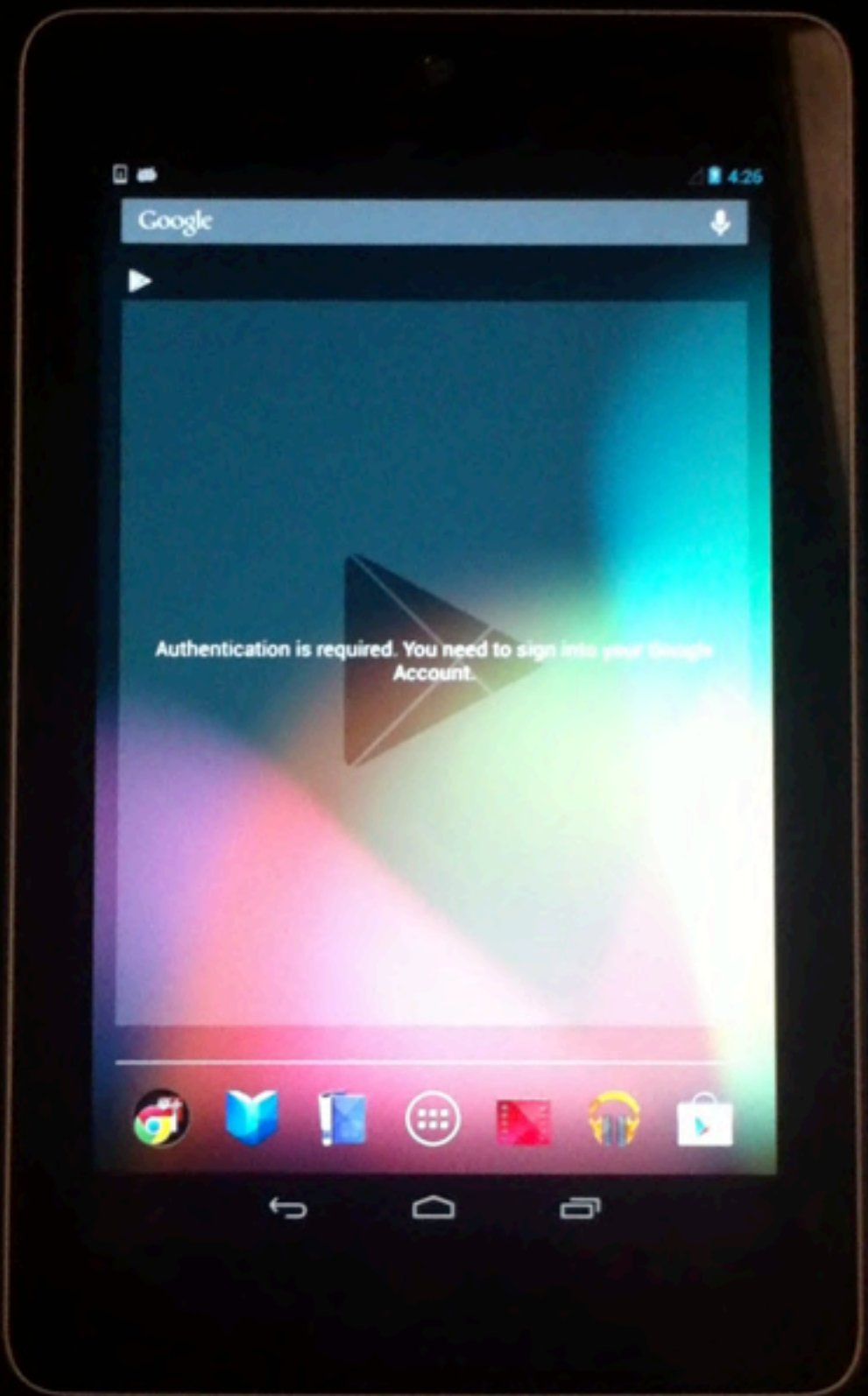
~/C/appium-jlipps git:master >>>

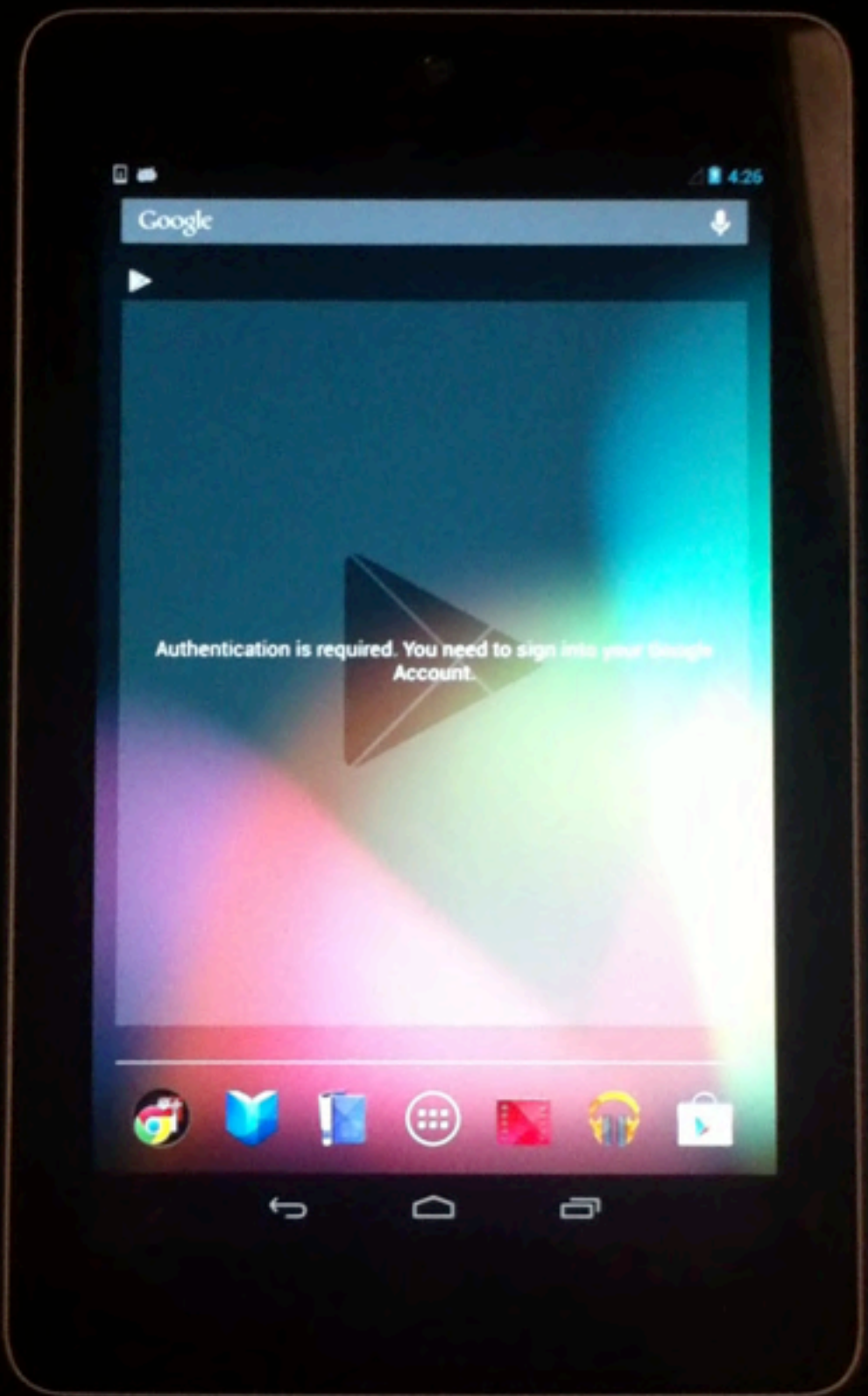
...xamples/node

~/C/a/s/e/node git:master >>> |

...ppium-jlipps

~/C/appium-jlipps git:master >>>



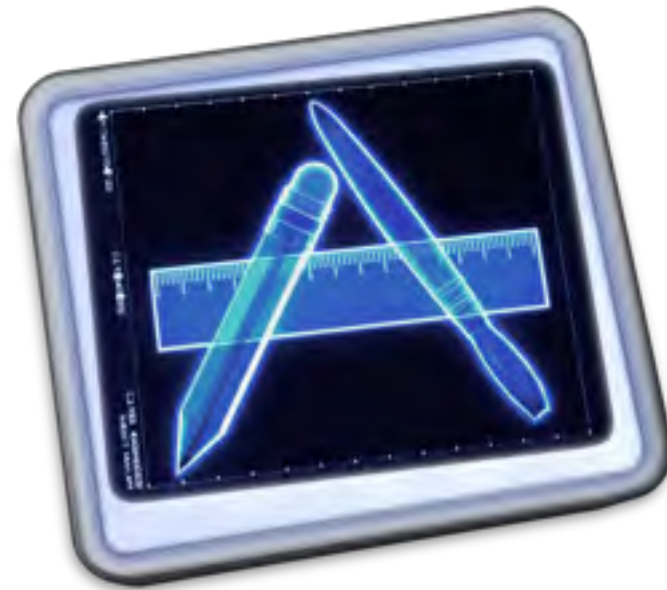


appium architecture



Automation Orchestra

Apple Instruments & UIAutomation for iOS



Automation Orchestra

Apple Instruments & UIAutomation for iOS

Google UiAutomator for Android (4.2.1 up)

uiautomator

The `uiautomator` testing framework lets you test your user interface (UI) efficiently by creating automated functional UI testcases that can be run against your app on one or more devices.

For more information on testing with the `uiautomator` framework, see [UI Testing](#).



Automation Orchestra

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

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

Google **Instrumentation** for older Android & hybrid
(via **Selendroid**)



Automation Orchestra

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

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

Google **Instrumentation** for older Android & hybrid
(via **Selendroid**)

Coming
Soon

Google **DroidDriver** for all Android versions



Automation Orchestra

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

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

Google **Instrumentation** for older Android & hybrid
(via **Selendroid**)

Coming
Soon

Google **DroidDriver** for all Android versions

Coming
Soon

Apple **XCUITest** for iOS 9+



Selenium WebDriver



is the globally-recognized standard
for browser automation



Selenium WebDriver is a HTTP API

POST /session

POST /session/element

GET /session/element/:id/:attr



Selenium WebDriver
is a W3C working draft



appium is a Node.js HTTP server
that creates and handles
WebDriver sessions



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



appium is working with the
Selenium project so we can
standardize these extensions



appium clients give access to these
extensions in many languages

Java : JS : Python : Ruby : PHP : C# :
ObjC : Perl

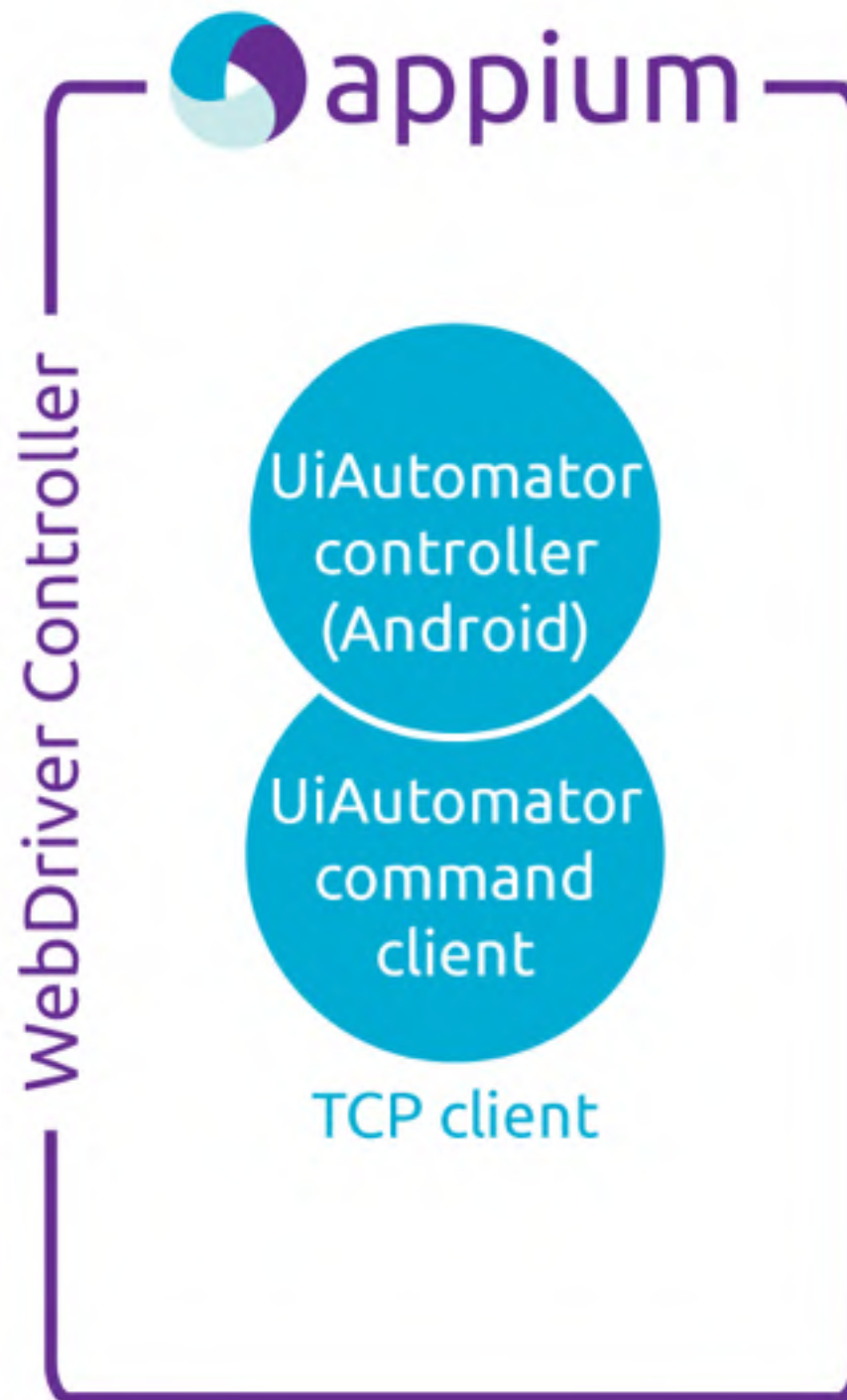


appium can run on your computer,
in your network, or on a cloud
service like **Sauce Labs**



Appium script

```
def test_required_fields
  self.get("signup")
  sbutton = self._id("s
  sbutton.submit()
  for field in [f['name
```



appium setup



Requirements

Basically the same as dev toolkits
for iOS, Android

Xcode + iOS SDK

Android SDK + Java



Install: Option One

Clone from GitHub

```
REPO="appium/appium.git"  
git clone https://github.com/\$REPO  
cd appium && ./reset.sh  
node .
```



Install: Option Two

Install from NPM

```
npm install -g appium  
appium
```

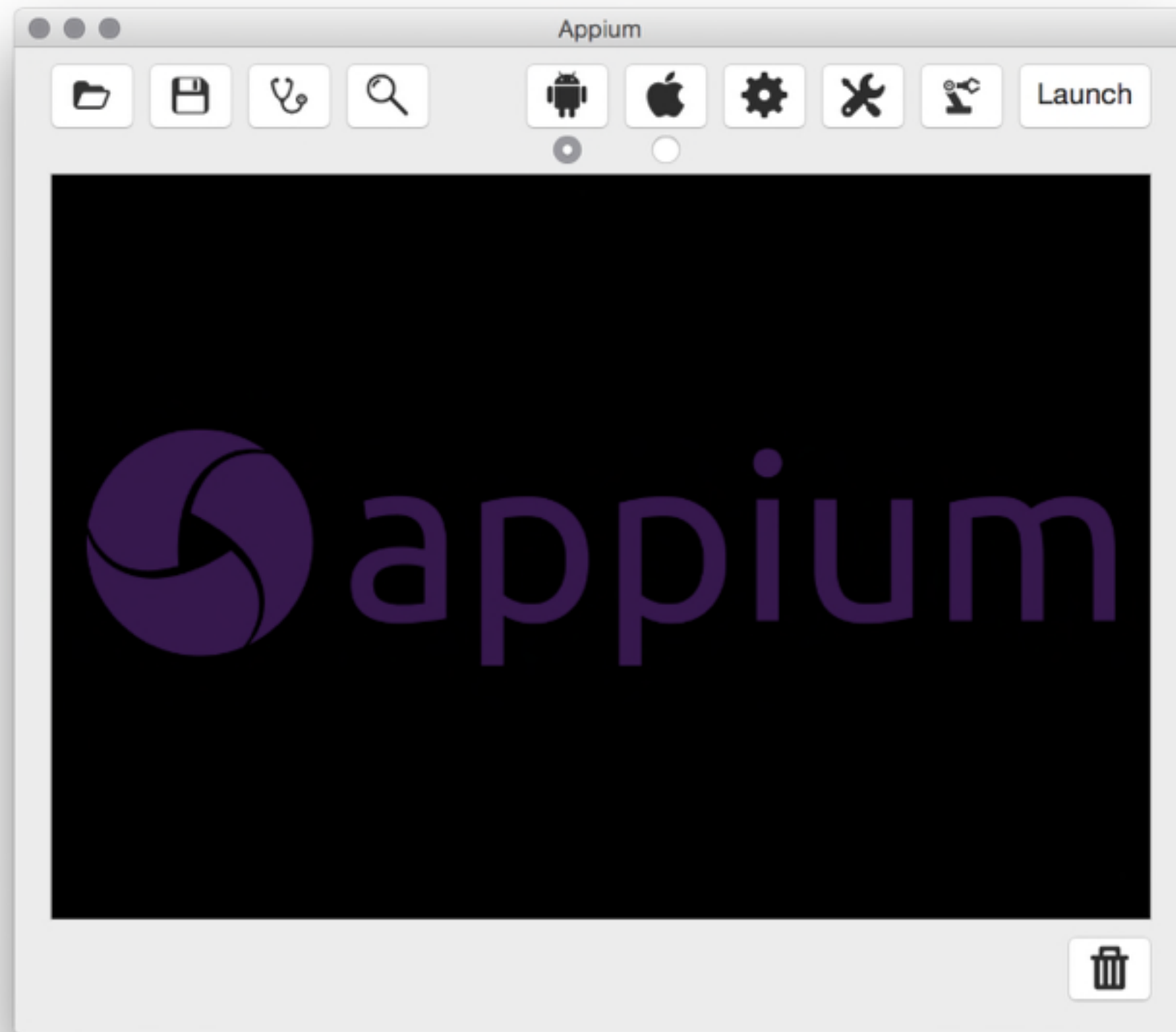


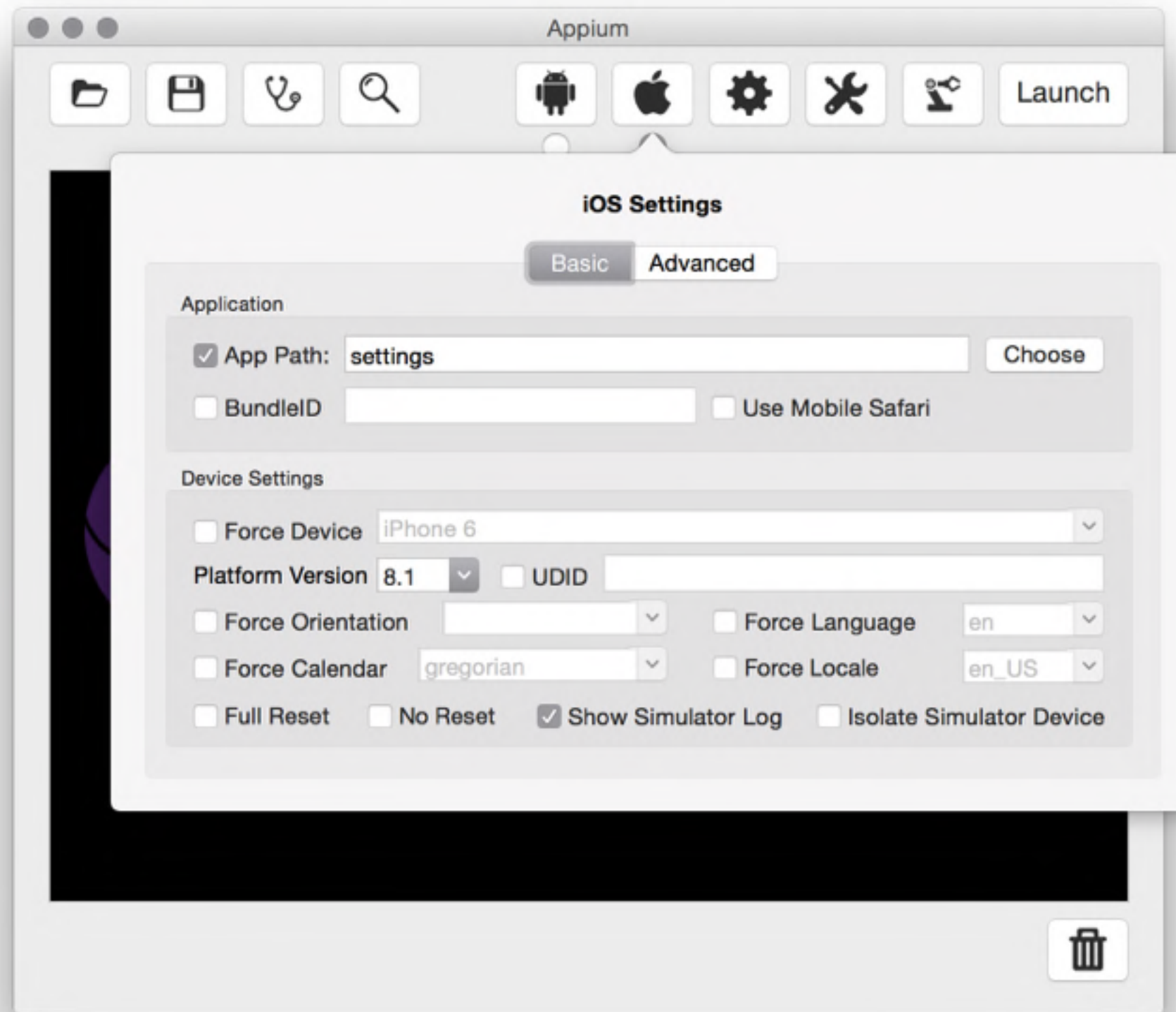
Install: Option Three

Download the GUI app

```
# github.com/appium/appium/releases  
open /Applications/Appium.app
```







Appium Inspector

Filters

Show Disabled Show Invisible

Record Refresh

Carrier 11:41 AM

Settings

[UIAWindow] [UITableView] rows [UITableViewCell] Genera

[UINavigationController] ▶ [UITableViewCell] G... ▶ [UIAStaticText]...

[UITableView] r... ▶ [UITableViewCell] Pr... ▶

[UIToolbar] ▶ [UITableViewCell] iC... ▶

[UITableViewCell] M... ▶

[UITableViewCell] S... ▶

[UITableViewCell] P... ▶

[UITableViewCell] G... ▶

[UITableViewCell] T... ▶

[UITableViewCell] F... ▶

[UITableViewCell] Flickr ▶

[UITableViewCell] Vi... ▶

[UITableViewCell] D... ▶

Details

name: General
type: UITableViewCell
value:
label: General
enabled: true
visible: false
valid: true
location: {0, 134}
size: {375, 44}
xpath: //
UIApplication[1]/
UIWindow[1]/
UITableView[1]/
UITableViewCell[1]

Context

no context

Change

Touch Text Locator Misc

Tap Swipe Shake

Precise Tap Scroll To

Copy XML 0°

General

Privacy

iCloud

Maps

Safari

Photos & Camera

Game Center

Twitter

Facebook

Flickr

Vimeo



No Install

Use Sauce Labs

<https://saucelabs.com/mobile>



appium test model



Desired Capabilities

```
DesiredCapabilities capabilities = new DesiredCapabilities();  
capabilities.setCapability("platformName", "iOS");  
capabilities.setCapability("deviceName", "iPhone Simulator");  
capabilities.setCapability("platformVersion", "7.1");  
capabilities.setCapability("browserName", "Safari");
```



Desired Capabilities

```
DesiredCapabilities capabilities = new DesiredCapabilities();  
capabilities.setCapability("platformName", "iOS");  
capabilities.setCapability("deviceName", "iPhone Simulator");  
capabilities.setCapability("platformVersion", "7.1");  
capabilities.setCapability("app", "/path/to/my.app");
```



Sessions

```
driver = new IOSDriver(new URL(appiumServer), capabilities);  
// driver = new AndroidDriver(new URL(appiumServer), capabilities);  
...  
driver.quit();
```



Finding Elements

```
import io.appium.java_client.MobileBy;  
driver.findElement(MobileBy.className("UIAButton"));  
driver.findElement(MobileBy.xpath("//android.widget.EditText"));  
driver.findElement(MobileBy.id("com.aut.android:id/searchButton"));  
driver.findElement(MobileBy.AccessibilityId("Sign In"));
```



Interacting with Elements

```
WebElement name = driver.findElement(MobileBy.AccessibilityId("username"));  
name.sendKeys("jlipps");
```

```
WebElement button = driver.findElement(MobileBy.className("UIAButton"));  
button.getText(); // "Sign In"  
button.click();
```



Interacting with Webviews

```
// do some native stuff
WebElement nativeButton = driver.findElement(By.className("UIAButton"));
nativeButton.click();

// switch to webview and do some webview stuff
driver.context("WEBVIEW_1");
WebElement search = driver.findElement(By.cssSelector(".search-key"));
search.sendKeys("j");

// go back to native
driver.context("NATIVE_APP");
```



Interacting with the Device

```
// device
driver.lockScreen(5);
driver.shake(); // iOS-only
driver.hideKeyboard();
driver.getOrientation();
driver.rotate(ScreenOrientation.LANDSCAPE);
driver.sendKeyEvent(82); // Android-only

// apps
driver.resetApp();
driver.closeApp();
driver.launchApp();

// geolocation
driver.location();
driver.setLocation(new Location(47.368650, 8.539183, 0));
```



Gestures

```
driver.swipe(150, 400, 150, 200, 800);  
driver.pinch(300, 450);  
driver.zoom(300, 450);
```

```
TouchAction dragNDrop = new TouchAction(driver).  
    longPress(e11).  
    moveTo(e12).  
    release();  
dragNDrop.perform();
```



Multiple devices (Android-only)

```
# get device IDs from `adb devices`  
  
appium --port 4730 --device-port 4731 -U $DEVICE1_ID  
appium --port 4740 --device-port 4741 -U $DEVICE2_ID  
  
# now configure your tests to have configurable ports  
  
APPIUM_PORT=4730 mvn test # run on first device  
APPIUM_PORT=4740 mvn test # run on second device
```



appium scale



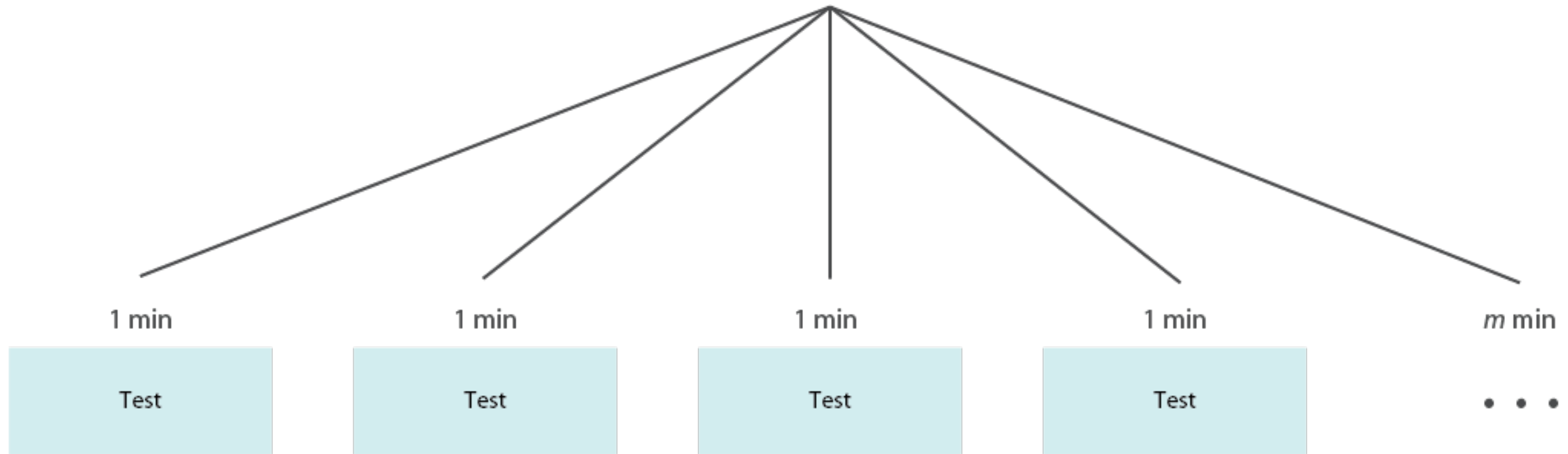
appium is great for local test
development, but has limitations when
scaling up for use in CI



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



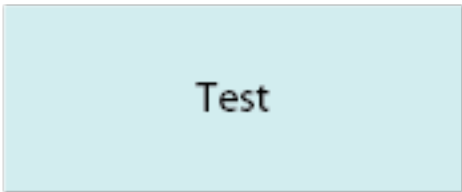
$n \rightarrow \infty$ min



1 min

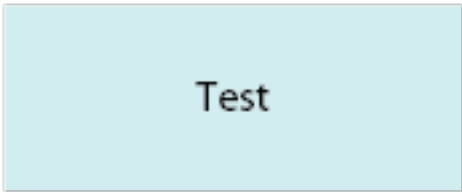
$n \rightarrow \infty$
concurrency

1 min



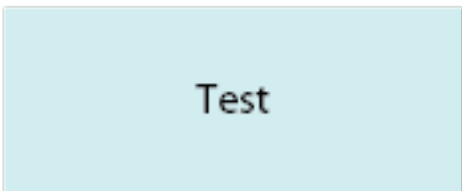
Test

1 min



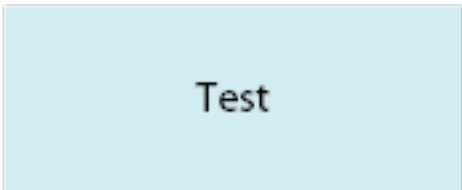
Test

1 min



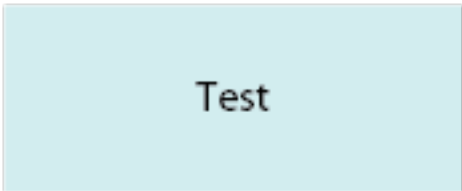
Test

1 min



Test

1 min



Test

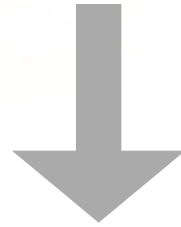
1 min

•
•
•



Running tests on Sauce

```
String appiumServer = "http://localhost:4723/wd/hub";  
driver = new IOSDriver(new URL(appiumServer), capabilities);
```



```
String username = "jlipps"  
String accessKey = "xxxx-xxxx-xxxx-xxxx"  
String appiumServer = MessageFormat.format(  
    "http://{0}:{1}@ondemand.sauce labs.com:80/wd/hub",  
    username, accessKey);  
driver = new IOSDriver(new URL(appiumServer), capabilities);
```



Dashboard

Tunnels

Archives

Docs

Test Success

Basic ios native test

Team

Report

Delete

Admin

Virtual Machine

iPhone 8.2

TestApp 7.1

Sauce Connect Disabled

Build [Learn more about annotating tests](#)

Owner [jlipps](#)

Started Jun 19, 2015 at 7:04AM

Ended Jun 19, 2015 at 7:06AM

Duration 1m 51s

Watch

Commands

Logs

Metadata

FILTER: Command Has Screenshot

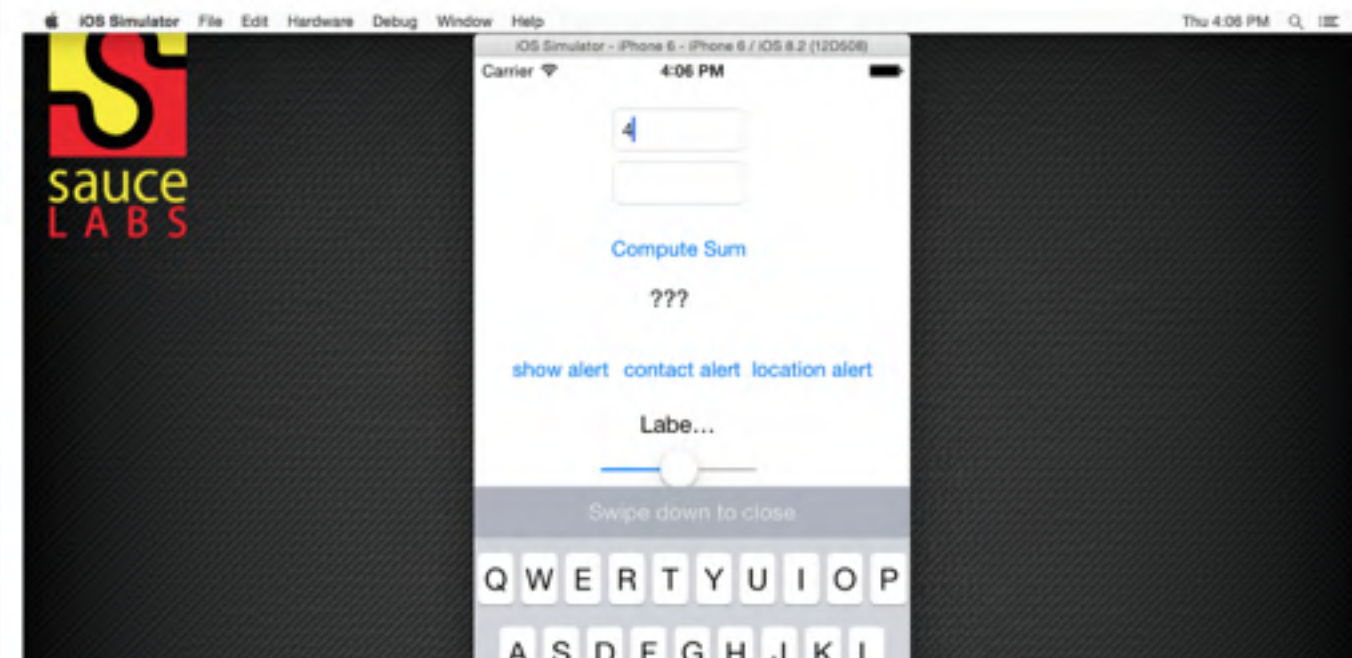
Play 4 of 10

00:00:32 00:00:38

POST	timeouts/implicit_wait	19.95ms	<input type="checkbox"/>
POST	elements	144.62ms	<input type="checkbox"/>
POST	element/0/value	2s	<input type="checkbox"/>
POST	element/1/value	1s	<input type="checkbox"/>
POST	element	196.89ms	<input type="checkbox"/>
POST	element/2/click	264.82ms	<input type="checkbox"/>
POST	element	146.1ms	<input type="checkbox"/>
GET	element/3/text	107.12ms	<input type="checkbox"/>
DELETE	/session/a7a2f3a7e195492cb6cee6a6154bb26b	0.06ms	<input type="checkbox"/>

0 Concurrent VMs

jlipps



Sauce Labs simulator and emulator
VMs are destroyed after each session



Sauce Connect allows your tests to run securely with all data transmitted through an encrypted private tunnel



 saucelabs.com

FREE FOR OPEN SOURCE



 CMTS2015

PROMO CODE FOR FREE TESTING





Android Real Devices





**Android
Real
Devices
Publicly
Available
...*soon***



appium future



Appium 1.5



```
Selendroid.prototype.insertSelendroidManifest = function (serverPath, cb) {
  logger.debug("Inserting selendroid manifest");
  var newServerPath = this.selendroidServerPath
    , newPackage = this.args.appPackage + '.selendroid'
    , srcManifest = path.resolve(__dirname, '..', '..', '..', 'build',
      'selendroid', 'AndroidManifest.xml')
    , dstDir = path.resolve(this.args.tmpDir, this.args.appPackage)
    , dstManifest = path.resolve(dstDir, 'AndroidManifest.xml');

  try {
    fs.mkdirSync(dstDir);
  } catch (e) {
    if (e.message.indexOf("EEXIST") === -1) {
      throw e;
    }
  }
  fs.writeFileSync(dstManifest, fs.readFileSync(srcManifest, "utf8"), "utf8");
  async.series([
    function (cb) { mkdirp(dstDir, cb); }.bind(this),
    function (cb) { this.adb.checkSdkBinaryPresent("aapt", cb); }.bind(this),
    function (cb) {
      this.adb.compileManifest(dstManifest, newPackage,
        this.args.appPackage, cb);
    }.bind(this),
    function (cb) {
      this.adb.insertManifest(dstManifest, serverPath,
        newServerPath, cb);
    }.bind(this)
  ], cb);
};
```



```
async buildNewModServer () {
  logger.info(`Repackaging selendroid for ${this.appPackage}`);
  let packageTmpDir = path.resolve(this.tmpDir, this.appPackage);
  let newManifestPath = path.resolve(this.tmpDir, 'AndroidManifest.xml');
  logger.info(`Creating new manifest`);
  await util.mkdirp(packageTmpDir);
  await writeFile(newManifestPath, await readFile(SE_MANIFEST_PATH, "utf8"));
  await this.adb.initAapt(); // TODO this should be internal to adb
  await this.adb.compileManifest(newManifestPath, this.modServerPkg,
                                this.appPackage);
  await this.adb.insertManifest(newManifestPath, SE_APK_PATH,
                                this.modServerPath);
  logger.info(`Repackaged selendroid ready at ${this.modServerPath}`);
}
```



[http://github.com/appium/appium/
milestones](http://github.com/appium/appium/milestones)



Issues

Pull requests

Labels

Milestones

New milestone

7 Open 7 Closed

Sort

Appium 1.5

No due date Last updated about 17 hours ago

0% complete 11 open 0 closed

Edit Close Delete

Appium 1.4

No due date Last updated about 10 hours ago

30% complete 72 open 31 closed

Edit Close Delete

Appium 1.3.4

No due date Last updated about 1 hour ago
bugfix release

0% complete 6 open 0 closed

Edit Close Delete

Appium 2.0

No due date Last updated about 17 hours ago
Stuff for a long way off that we eventually want to get to

6% complete 15 open 1 closed

Edit Close Delete

Appium 3.0

No due date Last updated about 1 hour ago

0% complete 1 open 0 closed



The future is in your hands...



- Write code, be a committer
- Make our docs better
- Share your experiences
- Help with community support
- Let us know what to focus on
- Keep up the translations!



Questions?



<http://appium.io>

<https://github.com/appium/appium>

@AppiumDevs • @jlipps • @saucelabs

谢谢

Thanks!



<http://appium.io>

<https://github.com/appium/appium>

@AppiumDevs • @jlipps • @saucelabs