







appium

By Lokesh Soni







#### Lokesh Soni

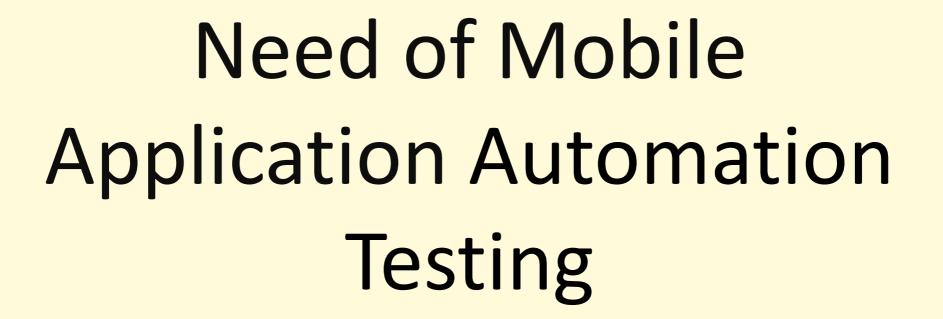
- Software Test Engineer @ Blisstering Solutions
- Web and Mobile Automation Tester
- Also, Manual Test Engineer

















Smartphone Apps have experienced explosive growth since 2007

> Mobile App are more reliable for Organizations

> Mobile Apps bringing half of their Web Traffic

In some cases, mobile apps are the only way customers interact with their products.



➢ No Wonder, that Mobile Apps Development Process has become quick and efficient, which also have high priority for organizations.

➢ Fortunately, Mobile development tools are evolving to meet these needs, with modern dev teams.

Development of Mobile Apps has become faster

But, what about the Testing approaches??





#### **Testing Mobile Applications is:**

More complex
 Time consuming
 Platform Variations
 Quality Concerns

So, alike Manual Mobile Testing process, we should also adopt **Mobile Automation Testing** 



# Why Mobile Automation Testing?





Efficient
Faster
Reusability
Reliable
Cost Reduction



# Which Mobile Automation Tool to be Used?







# Appium



# What is *Appium*? & Why *Appium*?



- Appium is an open-source test automation tool
- Allows testing for all types of Mobile Applications: Native Apps, Hybrid Apps and Mobile Web Apps
- Importantly, it is "Cross-Platform"







iOS	Android	
calabash-ios Frank UIAutomation ios-driver KeepItFunctional	calabash-android MonkeyTalk Robotium UiAutomator selendroid	

Sappium











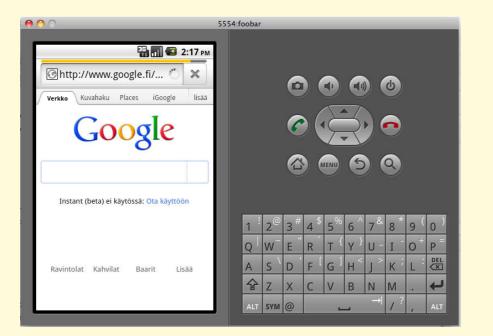


#### **Appium** supports most of the Programming languages like java, ruby, php, C#, etc.





# **Appium** handles both, Simulators and Real devices





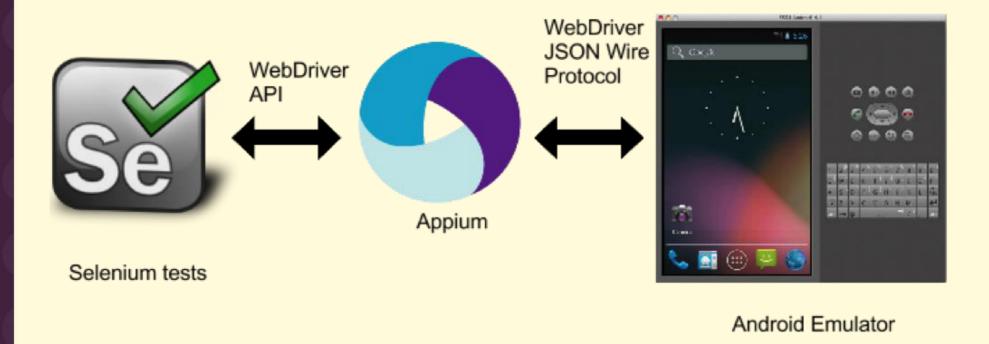


# **Appium** supports Automation for all Apps like Hybrid, Native and Web apps





# **Appium** is compatible with Selenium Webdriver









#### **Appium** is free and Open source

#### Appium is very well supported and Active Google group, Building a large and thriving open source community effort







### Getting Started with Appium





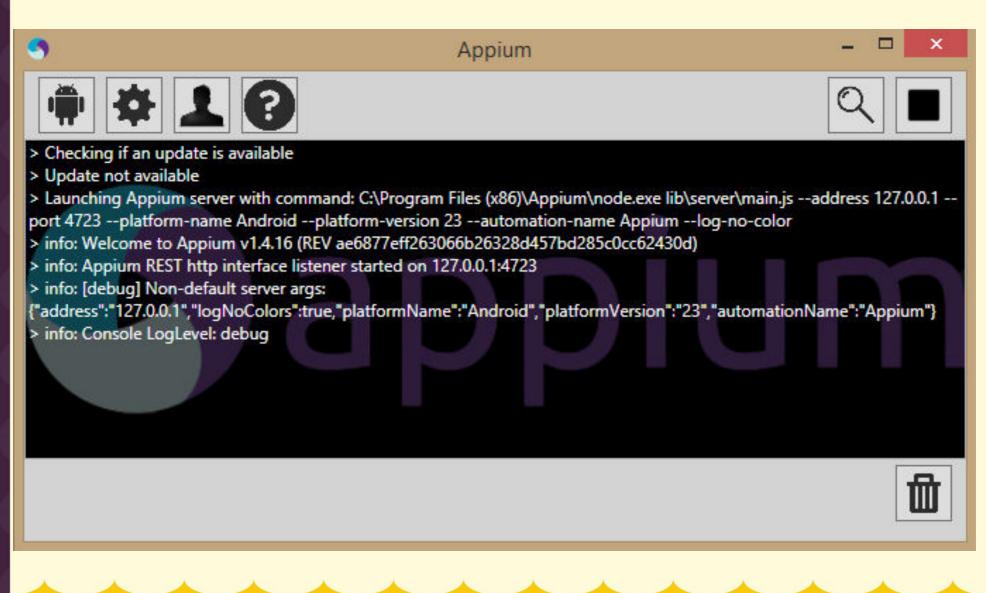
- JDK (Java Development Kit) V1.6(+)
- Eclipse Indigo(+)
- Android SDK With API level 17(+)
- Selenium WebDriver JAR 2.43.0(+)
- Appium for Mac/Windows 1.3.7.2(+)
- APK App info from Play Store







#### **Appium Screenshot**







#### **UI Automator**

0 0	Ap	pium Inspector	
Filters			
Show Invisible Sho	w Disabled	Record Refresh	2/ B 956
	[view]	[window]	Accessibility
			Animation App
[view]		[list]	Content
[window]	[window]	•	Graphics
			Media
			NFC
			os
			Preference
			Text
			Views
			Details C 0°
Touch Tap Swipe Sh Precise Tap Scroll 1	Text Misc ake		content-desc: Accessibility class: android.widget.TextView text: Accessibility index: 0 enabled: true clickable: true location: {0, 108} size: {800, 64} xpath: //view[1]/window[2]/list[1]/ text[1]







#### **Test Script Screenshot (On Eclipse)**

#### WebDriver driver: @BeforeClass public void setUp() throws MalformedURLException{ String Device = "Nexus 5"; String DeviceVersion = "6.0"; DesiredCapabilities capabilities = new DesiredCapabilities(); capabilities.setCapability("deviceName", Device); capabilities.setCapability("appium-version", "1.3.4.1"); capabilities.setCapability("platformVersion", DeviceVersion); capabilities.setCapability("platformName", "Android"); capabilities.setCapability("appPackage", "com.bliss.Twit.Tv"); // This is package name of your app (you can get it from apk info app) capabilities.setCapability("appActivity", "com.bliss.Twit.Tv.TwitActivity"); // This is Launcher activity of your app (you can get it from apk info app) //Create RemoteWebDriver instance and connect to the Appium server. //It will launch the App in Android Device using the configurations specified in Desi driver = new RemoteWebDriver(new URL("http://127.0.0.1:4723/wd/hub"), capabilities); driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS); 3





### How Appium works?

Sappium Working Architecture



#### Working Architecture of Appium on Real Devices/Simulator





#### Working flow:

1. From Web-driver, Automation Commands are sent in form of JSON via HTTP request to Appium Server.

2. Appium Server invokes Vendor specific mechanism to execute those commands on the Mobile-Device.

3. Client sends back the message to the Appium Server.

4. Appium Server logs the result in the console of the Web Driver.







### Demo of Appium Automation Script





#### **Limitations of Appium:**

- Doesn't support image comparison
- Doesn't support testing of Android Version lower than 4.2
- Limited support for testing Hybrid App
- Long time to configure Appium for both android and iOS
- No Support to run Appium Inspector on Microsoft Windows







# **Any Question?**







#### WHAT DID YOU THINK? EVALUATE THIS SESSION

asia2016.drupal.org/schedule