

STAR(テスト自動化研究会) 第18回研究会

Appiumについて

2015.1.18

@sumio_tym (TOYAMA Sumio)

アジェンダ

- ・ Appiumの概要
- ・ セットアップから実行まで
- ・ テストスクリプトの書き方
- ・ インспекタの使い方
- ・ まとめ

Appiumの概要

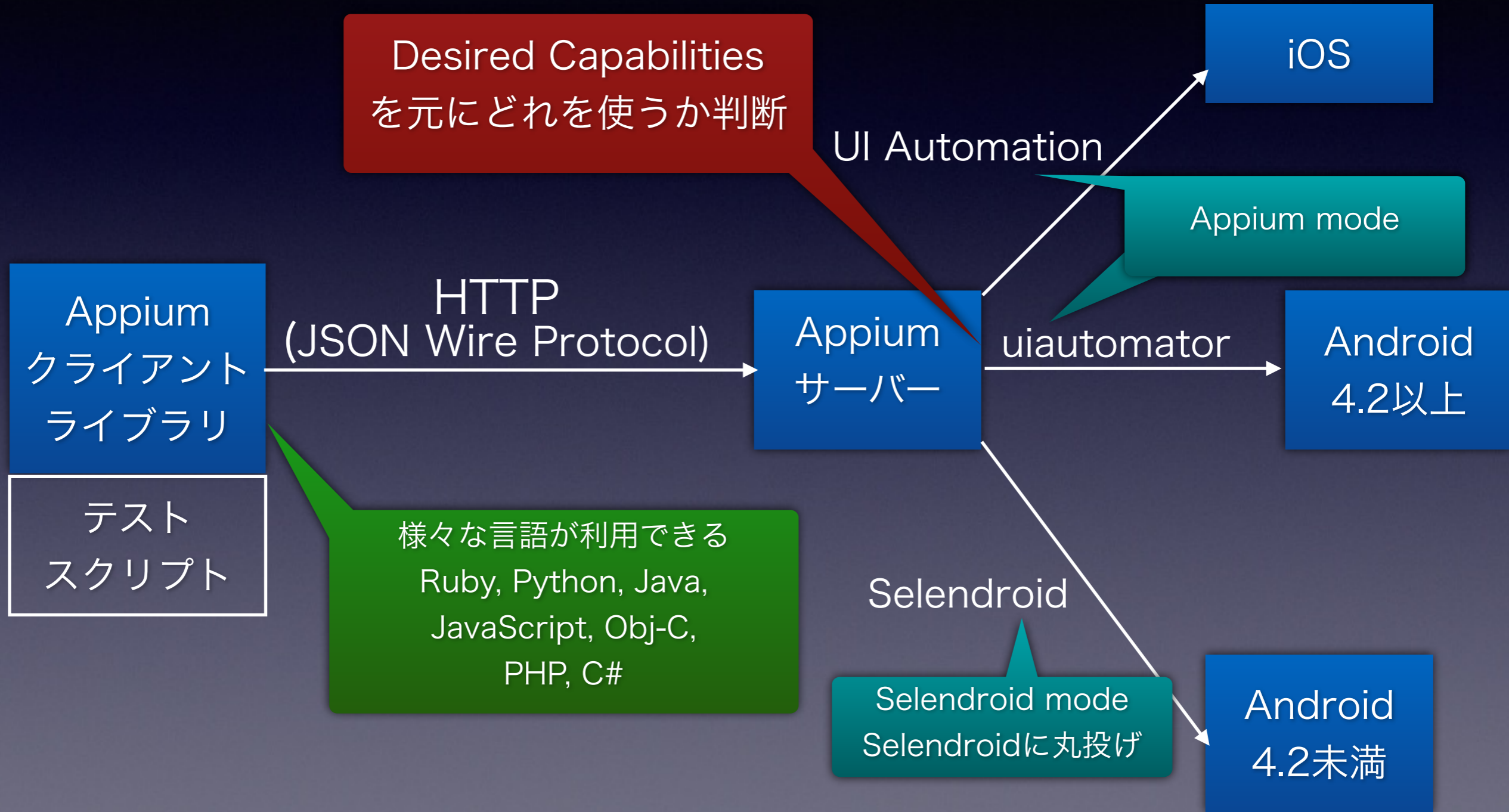
Appiumの概要

- ホームページ: <http://appium.io/>
- github: <https://github.com/appium/appium>
- ライセンス: Apache License, Version 2.0
- スポンサー: Sauce Labs社

Appiumの特徴

- Android/iOS両対応
 - Android 2.3以上、iOS 6.0以上
 - ネイティブ/ハイブリッド(WebView)両対応
- WebDriver準拠
 - Selenium2ユーザーには学習コスト低め

アーキテクチャ



セットアップから
テスト実行まで

Appium Desktop Apps

- ・ 「Appiumサーバー」の起動・終了をGUIで操作できる
 - ・ [Launch]ボタンでサーバー起動
- ・ インспекタ機能あり
- ・ Windows/Mac OS X対応
- ・ <https://bitbucket.org/appium/appium.app/downloads/> からダウンロード
- ・ インストールは特に難しいところなし

クライアントライブラリ

- Java / JUnit4を使う場合

```
apply plugin: 'java'
```

```
repositories {  
    mavenCentral()  
}
```

```
dependencies {  
    // Appiumクライアントライブラリを利用するための宣言  
    testCompile 'io.appium:java-client:2.1.0'  
}
```

テスト実行

- ・ 通常のJUnit4テストケース実行と同じ
 - ・ Eclipseから [Run As...]
 - ・ gradle test

テストスクリプトの 書き方

(JavaでAndroidの試験をする場合)

基本的な考え方

- Selenium Client Driverを拡張。
基本的な考え方は同じ:
 - Desired Capabilitiesの指定
 - findElement(<ロケータ>)でUI部品を探す
 - 見付かったらその部品を操作・検証する
- APIドキュメント
 - <http://appium.github.io/java-client/>
 - <http://selenium.googlecode.com/git/docs/api/java/index.html>

Desired Capabilities

- 必須項目は3つ
 - browserName: ""を指定
 - DeviceName: 無視される。何でもOK
 - app: 試験対象apkへの絶対パス
- その他重要なもの
 - automationName: Selendroid modeにしたいとき
 - unicodeKeyboard: 日本語など入力したいとき
 - etc.
- 詳細は:

<http://appium.io/slate/en/v1.3.4/?java#appium-server-capabilities>

ロケータ

- Androidネイティブアプリの場合:
 - `By.className()`: Viewのクラス名
 - `By.xpath()`: XPath
 - `MobileBy.AccessibilityId()`: `contentDescription`属性
 - `MobileBy.AndroidUIAutomator()`: `uiautomator`
- ❖ 表示テキストを使う素直な方法がない
- ❖ XPath構築はインスペクタの助けが必要
- ❖ Selendroid modeは別ルール
<http://selendroid.io/native.html>
- ❖ WebViewの中も別ルール

デモ

インスペクタ

まとめ

注意点

- ・ テスト対象アプリの改造が必要なケースあり
 - ・ Selendroid mode時のINTERNETパーミッション
 - ・ Appium modeでWebViewの試験
`WebView.setWebContentsDebuggingEnabled(true);`
- ・ Appium 1.0を境に大きな仕様変更あり
 - ・ 古い記事やドキュメントに注意!

所感

- ・ これだけでAndroid/iOS × ネイティブ/WebViewに対応できるのはすごい。
 - ・ プラットフォームごとにロケータの意味が違うので、スク립ト共通化にはPage Object Patternが必須
- ・ 開発が活発
 - ドキュメントは追いついていない…。
 - バグはそれなりにある
- ・ Selenium2の経験がないとやや難しい
 - 「部品が表示されるまで待つ」とか

おわり