





CALABASH

Angelia Wang
Aug 5, 2013

- WHAT CALABASH DOES
- HOW IT WORKS
- GETTING STARTED
- CROSS-PLATFORM PRACTICES
- X-PLATFORM EXAMPLE
- CALABASH FOR X PROJECT

What Calabash does?

- Brings Cucumber to mobile native and hybrid apps.
- Functional testing for mobile apps on both devices and simulators.
- Calabash consists of two libraries calabash-android for testing android app and calabash-iOS for testing iOS app.

How Calabash works?

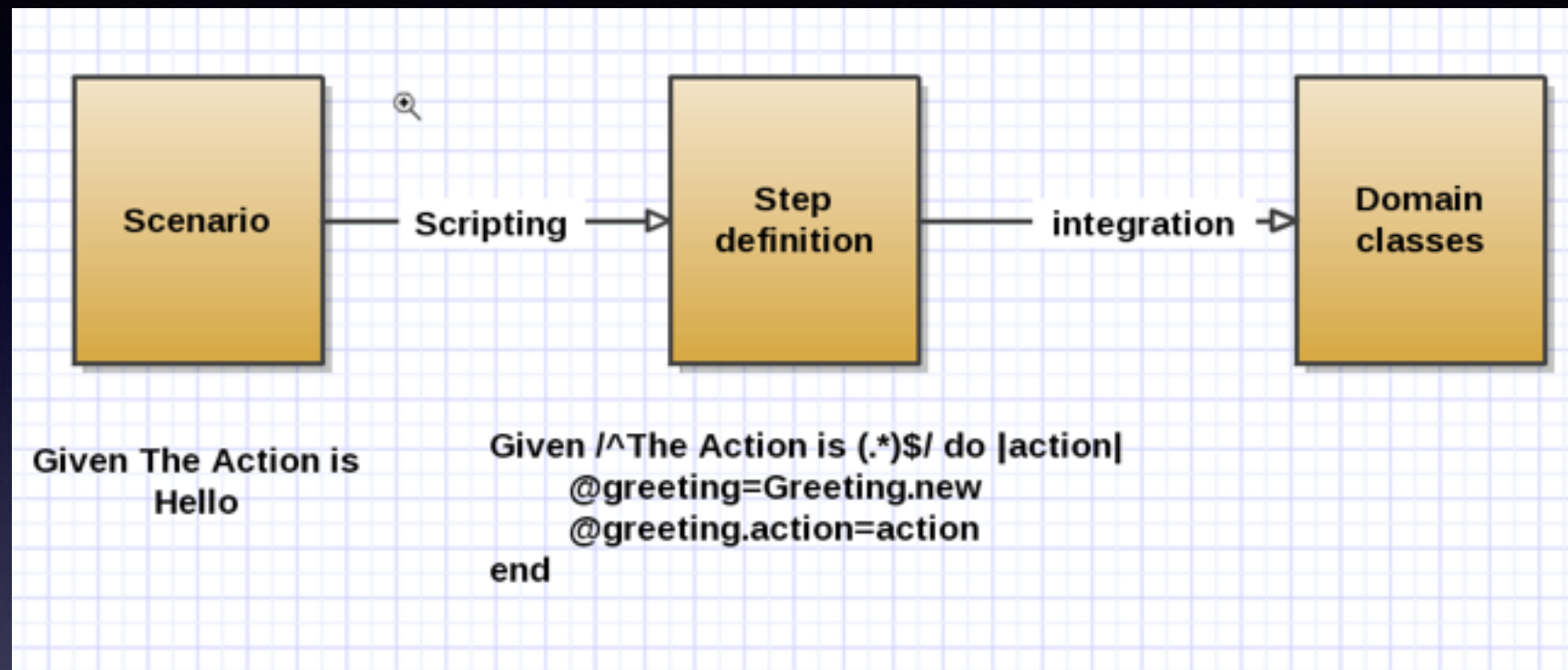
- Based on Cucumber
- Details, see next slides...

Cucumber



- A tool that executes plain-text functional descriptions as automated tests.
- The language that Cucumber understands is called Gherkin.

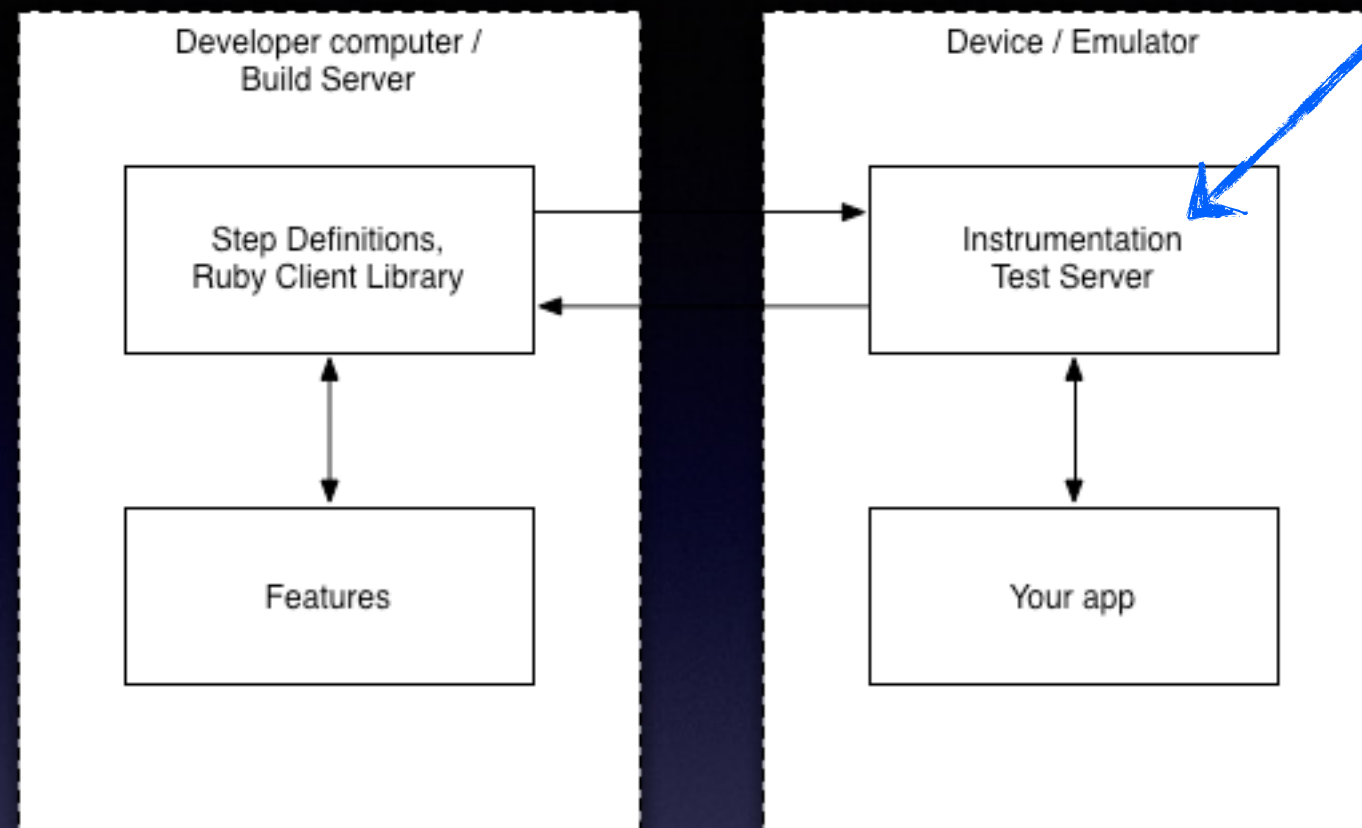
Cucumber-Gherkin



- Features: describes behavior of the app.
- Scenario: capture one key example for a feature.
- Step Definition: predefined, customized

Calabash-android





- Instrumentation Test Server: another app that will be installed and executed on the device.
- you don't need to make modification to your app before testing.

- **ActivityInstrumentationTestCase2** : provides functional testing of a single activity.

```
public abstract class
ActivityInstrumentationTestCase2
extends ActivityTestCase

java.lang.Object
↳ junit.framework.Assert
    ↳ junit.framework.TestCase
        ↳ android.test.InstrumentationTestCase
            ↳ android.test.ActivityTestCase
                ↳ android.test.ActivityInstrumentationTestCase2<T extends android.app.Activity>
```

<http://developer.android.com/reference/android/test/ActivityInstrumentationTestCase2.html>

- Cucumber skeleton

```
features
|_support
| |_app_installation_hooks.rb
| |_app_life_cycle_hooks.rb
| |_env.rb
|_step_definitions
| |_calabash_steps.rb
|_my_first.feature
```

- [Run tests] calabash-android run <apk>
- [Console] calabash-android console <apk>

Ruby API

- `start_test_server_in_background`
- `reinstall_apps`
- `Query`
- ...

https://github.com/calabash/calabash-android/blob/master/documentation/ruby_api.md

Query

- it works on calabash-android 0.4.0.pre2+
- classname
- direction
- filtering

```
def change_language(language)
  language.lstrip!
  touch("* marked:'btnLanguage'")
  sleep(STEP_PAUSE)
  wait_for_elements_exist(["* id:'listLanguage'"])
  wait_for {
    performAction('scroll_down') if !element_exists("* text:'#{language}')"
    element_exists("* text:'#{language}')"
  }
  touch("* text:'#{language}')"
  sleep(STEP_PAUSE)
  $language = language
end
```

```
def login_enabled?
  query("* marked:'btnLogin').first["enabled"] == true
end

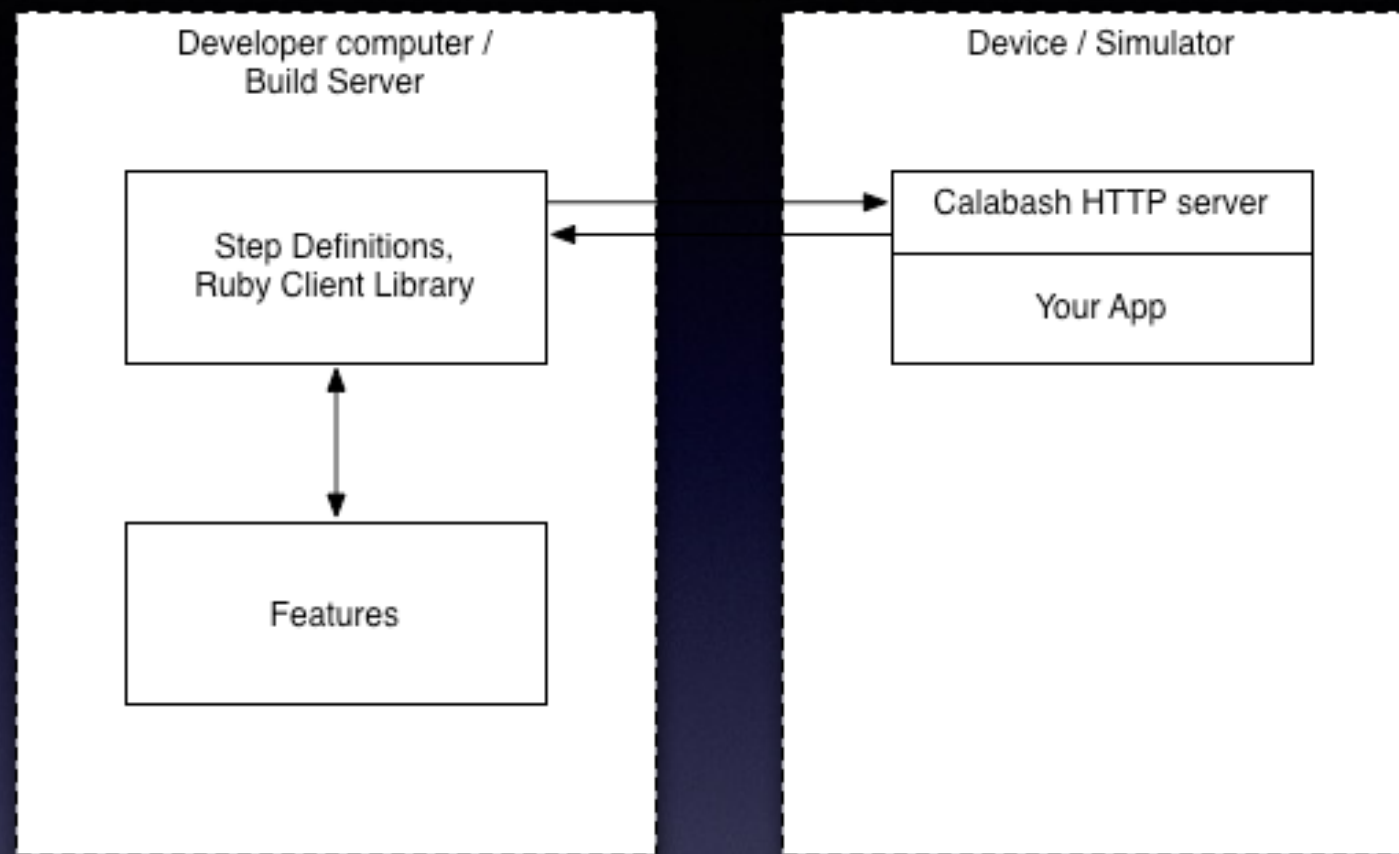
def login_failed?
  !q("* id:'txtAlertTitle').empty? && q("* id:'common_loading_view').empty?
end
```

WebView Support

- Calabash Android supports querying and acting on webview content
- query, touch, entering text
- e.g. query("webView css:'#header'"), query("webView css:*")...

Calabash-iOS





- client library (written in Ruby)
- calabash.framework, server framework (written in Object-C)

- Cucumber skeleton

```
features/  
├─ my_first.feature  
├─ step_definitions  
└─ support
```

- [Run tests] cucumber
- [Console] calabash-ios console

calabash-ios console

- it could help you explore your app interactively.
- e.g. `query("button")`, `touch("button index: 0")`, etc
- set accessibility ids and labels

Set accessibility ids

- on iOS5+ recommended approach is to use `accessibilityIdentifier`.
- e.g.

```
[languageButton  
setAccessibilityIdentifier:@"change  
language"];
```

Getting Started

- Android/iOS IDE&SDK
- Install Calabash-Android/iOS
- Install ImageMagick
- Cross-Platform Acceptance Testing Best Practices
- X-platform-example

Install Calabash-Android

- `sudo gem install calabash-android`
- `sudo gem install calabash-android -v 0.4.4`
- [Uninstall] `gem uninstall calabash-android`

Install Calabash-iOS

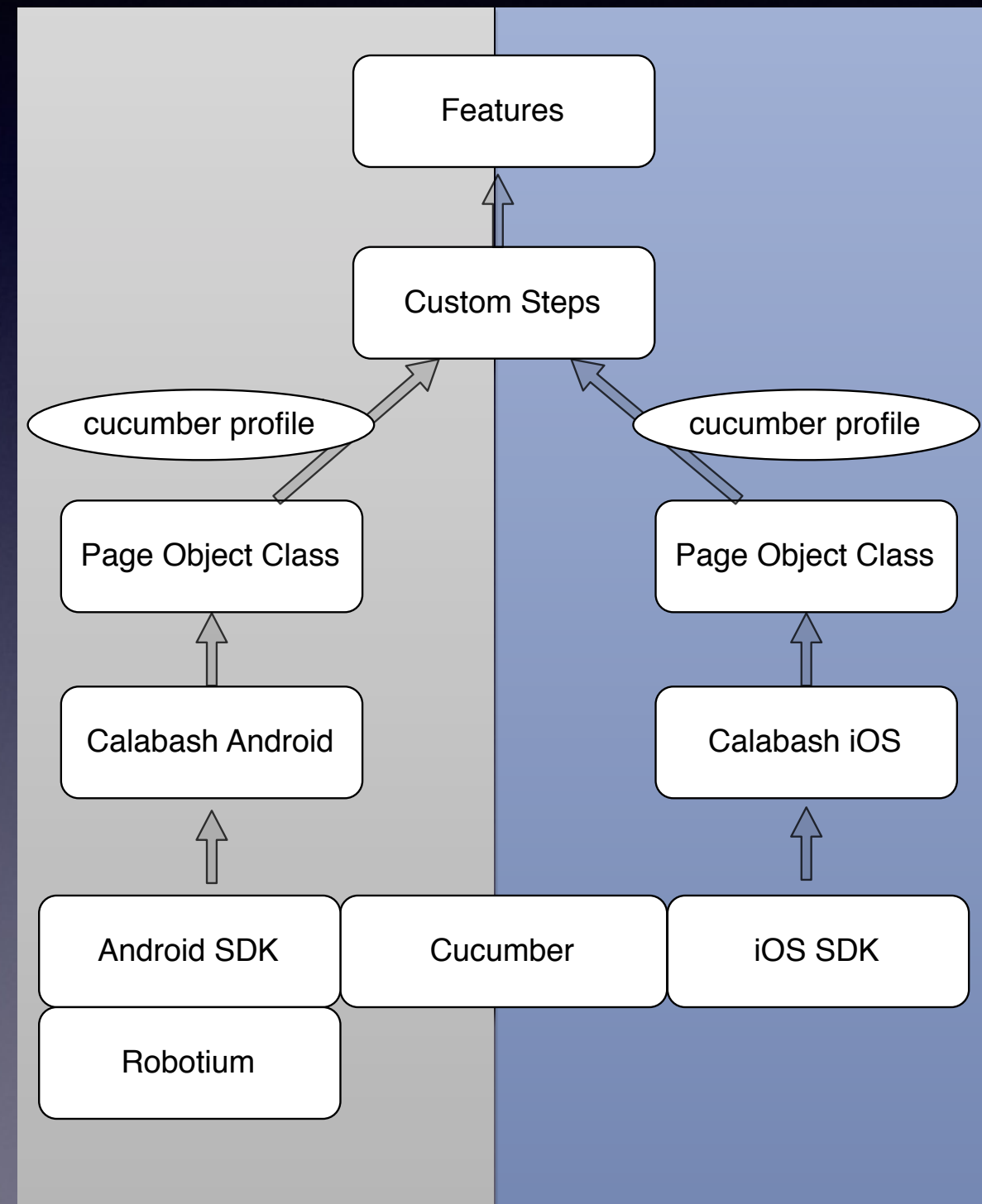
- `sudo gem install calabash-cucumber`
- `calabash-ios setup`
- `gem list calabash-cucumber //check current client library`

Install ImageMagick

- brew install imagemagick
- ImageMagick: a software suite to create,edit,compose or convert bitmap images.
- <http://www.imagemagick.org/script/index.php>

Cross-Platform Acceptance Testing Best Practices

- Page-Object Pattern (POP)
- Page Objects: abstract a single screen
- Cross-platform Core Idea
- Cross-platform in practice



X-platform-example

- Automate each screen
- Use only custom steps which only use page objects
- For each platform, define a class with implementations of the page-object methods
- Cucumber profile (config/cucumber.yml) for each platform

<https://github.com/calabash/x-platform-example>

Cucumber.yml

- Cucumber lets you store and reuse commonly used cucumber command line arguments for a project in a cucumber.yml/yaml file.
- This file must be in current working directory.
- you could see cucumber command lines by “`cucumber -help`”
- <https://github.com/cucumber/cucumber/wiki/cucumber.yml>

Practice for X Project

Structure

- android:cucumber profile,page object class,calabash-android config,screenshot
- iOS:cucumber profile,page object class,calabash-ios config,screenshot,recordings
- step_definitions:custom steps
- support:users,environment,screenshot functions,etc.
- features: ...

cucumber.yml

- `x: PLATFORM=android -r features/support -r features/android/support -r features/android/helpers -r features/step_definitions -r features/android/pages/ -r features/englishtown --format html --out calabash-testresult.html features/x features/y`

DEMO

- Add features
- Run Testing:Android/iOS
- Set server's environment
- Add test accounts
- custom steps
- ...

Notes for Android

- You need to check the calabash-android version. Currently for X Project, we use 0.4.4. If you have installed calabash-android by online, please uninstall first and then install 0.4.4.
- Add `<uses-sdk android:targetSdkVersion="SDK_VERSION" />` to `AndroidManifest.xml`

Reference

- Calabash-Android <https://github.com/calabash/calabash-android>
- Calabash-iOS <https://github.com/calabash/calabash-ios>
- <https://github.com/calabash/x-platform-example>
- <http://blog.lesspainful.com/>
- Cucumber <http://cukes.info/>
- <https://github.com/cucumber/cucumber/wiki>

Thanks