



Fastlane as a Submodule

Issarapong

Growth Session #24 - May 16-17 2019

- An open source platform that simplifying mobile development and distribution.
- In the iOS Chapter, we're using fastlane for
 - Testing
 - Archiving
 - Deploying
 - Code signing synchronization
 - Utils
 - Pull request creation
 - Generation of release notes

Intro: iOS Template

- We've been working for iOS Template for over a year.
- Contains:
 - Xcode Project with customized build configuration and schemes.
 - Fastlane script.
 - Jenkinsfile.
- The iOS chapter has not worked on iOS Compass hard enough to finalize conventions yet:
 - Code style.
 - Design pattern.
 - Layers.
 - Utils

Most of the updated are automation scripts: ./fastlane and Jenkinsfile.

But:

- Each project has its own workflow.
- Jenkinsfile and fastlane scripts need to be updated to be able to deal with different workflows.



Xcode CLI tool wrapper



```
bundle exec fastlane build_for_testing
```

What does Fastlane actually do ?

```
set -o pipefail && env NSUnbufferedIO=YES xcodebuild -workspace ./RedPlanet.xcworkspace -scheme Red\
Planet -destination 'platform=iOS Simulator,id=40E1A9B0-F8CA-415D-BE90-CD2DEAE75A3C' -derivedDataPath
'/Users/dekablade01/Library/Developer/Xcode/DerivedData/RedPlanet-dqvkfwgikmoolgpariwsjvgszu' clean
build-for-testing | tee '/Users/dekablade01/Library/Logs/scan/RP Staging-Red Planet.log' | xcpretty --
report html --output '/Users/dekablade01/Repositories/rphl-ios-app/fastlane/test_output/report.html' --
report junit --output '/Users/dekablade01/Repositories/rphl-ios-app/fastlane/test_output/report.junit'
--report junit --output '/var/folders/zg/yzxntzbj7cx9qw9yp_8xprww0000gn/T/junit_report20190517-62695-
zbnpkt'
```

Fastlane



Xcode CLI

Xcode CLI is shipped with Xcode.app

Which means every single project uses Xcode CLI

Then Fastlane script should be also able to be used
by any iOS Project ?


Introducing Fastlane **Submodule**

Goals

- Split ``./fastlane`` out of the iOS Template:
 - Sometimes we do not start a project from scratch.
- Parameterization
 - CI will be able to do a lot more stuff by passing parameters into a lane.
 - Compare with other platform, CI can do stuffs using bash directly. If we want to share this script across projects, we need to add parameters to public lanes as much as possible.
- Lane Visibility adjustment

- iOS Features:
 - Building, Testing, Distributing, Code Signing synchronization, etc ..
 - Mostly already in iOS Template.
- Modularization:
 - Most of the things are in `Clfile`
 - Split them by use case.

Test Module



```
module TestModule

  def self.init(fastlane)
    @@fastlane = fastlane
  end

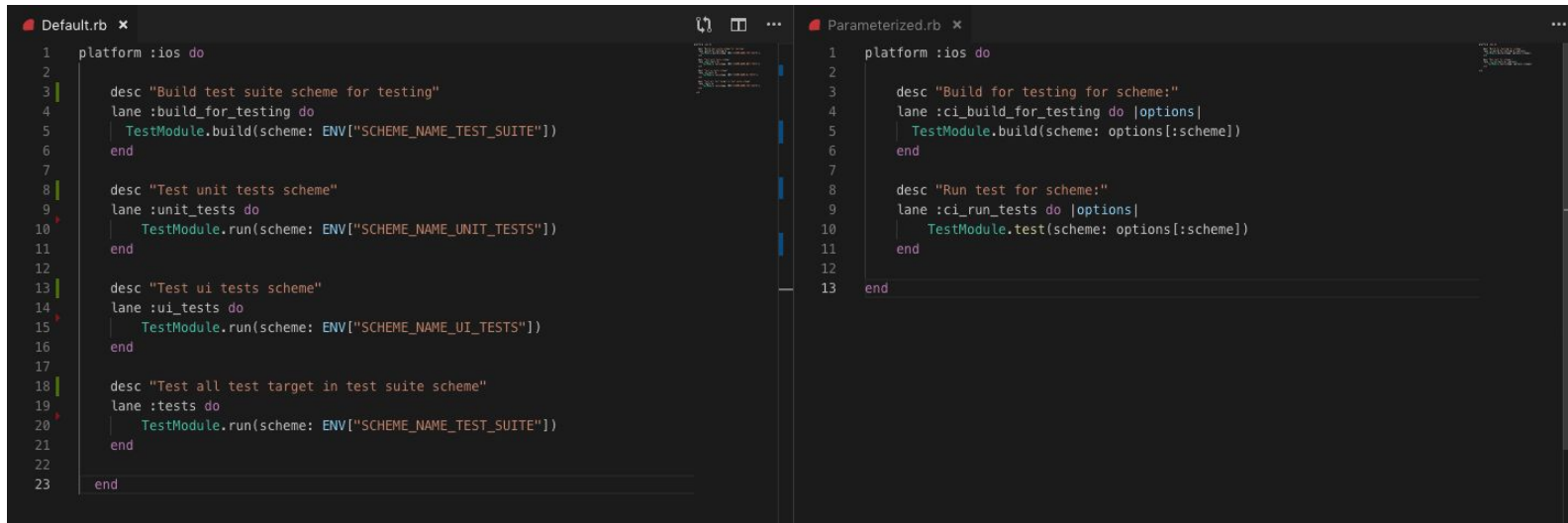
  def self.build(scheme:)
    @@fastlane.scan(scheme: scheme, build_for_testing: true)
  end

  def self.test(scheme:)
    @@fastlane.scan(scheme: scheme, test_without_building: true)
  end

end
```

Achievement and Progress

- Wrapper lanes for development machine.
- Parameterized lanes for CI machine.



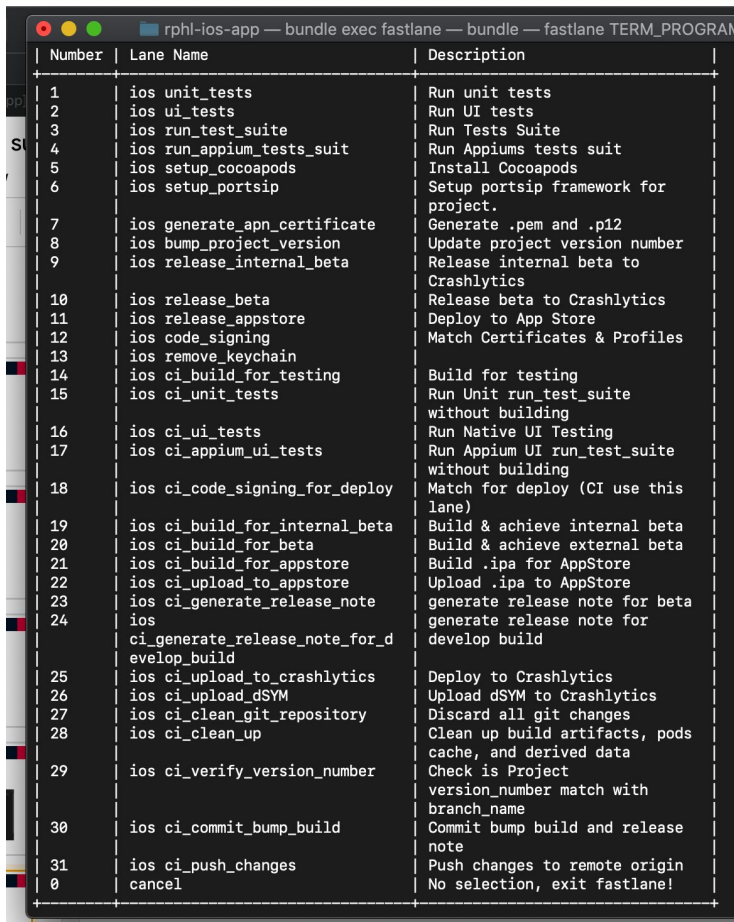
```
Default.rb x
1 platform :ios do
2
3   desc "Build test suite scheme for testing"
4   lane :build_for_testing do
5     TestModule.build(scheme: ENV["SCHEME_NAME_TEST_SUITE"])
6   end
7
8   desc "Test unit tests scheme"
9   lane :unit_tests do
10    TestModule.run(scheme: ENV["SCHEME_NAME_UNIT_TESTS"])
11  end
12
13  desc "Test ui tests scheme"
14  lane :ui_tests do
15    TestModule.run(scheme: ENV["SCHEME_NAME_UI_TESTS"])
16  end
17
18  desc "Test all test target in test suite scheme"
19  lane :tests do
20    TestModule.run(scheme: ENV["SCHEME_NAME_TEST_SUITE"])
21  end
22
23 end

Parameterized.rb x
1 platform :ios do
2
3   desc "Build for testing for scheme:"
4   lane :ci_build_for_testing do |options|
5     TestModule.build(scheme: options[:scheme])
6   end
7
8   desc "Run test for scheme:"
9   lane :ci_run_tests do |options|
10    TestModule.test(scheme: options[:scheme])
11  end
12
13 end
```

Lane's visibility

- High level lanes
- Low level lanes
- Example:
 - High level lane
 - `release_beta`
 - Low level lanes
 - `build_for_crashlytics`
 - `generate_release_notes`
 - `upload_to_crashlytics`
 - `upload_dSYM`

Achievement and Progress



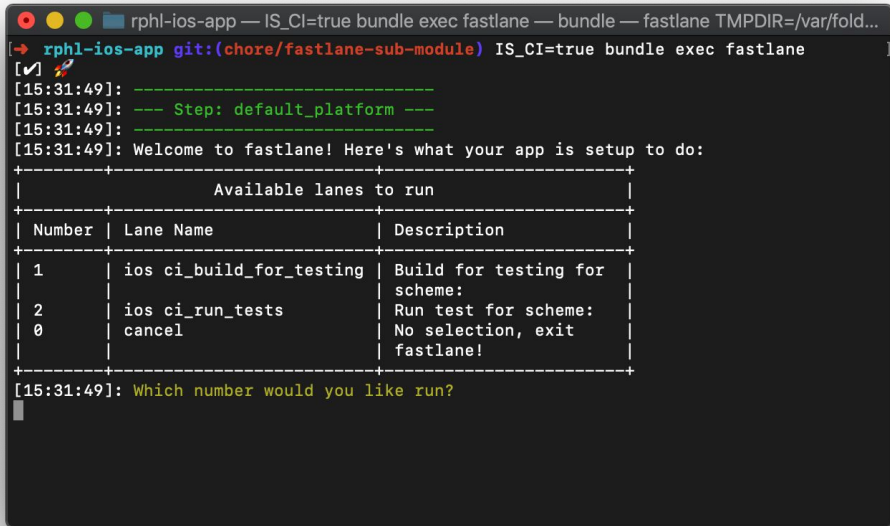
The screenshot shows a terminal window with a title bar that reads "rphl-ios-app — bundle exec fastlane — bundle — fastlane TERM_PROGRAM". The terminal displays a Fastlane lane configuration for "ios". The configuration is a table with three columns: "Number", "Lane Name", and "Description". The lanes are numbered 1 through 31, with a final "0" for "cancel". The descriptions provide details for each lane, such as running unit tests, setting up CocoaPods, generating certificates, and deploying to the App Store.

Number	Lane Name	Description
1	ios unit_tests	Run unit tests
2	ios ui_tests	Run UI tests
3	ios run_test_suite	Run Tests Suite
4	ios run_appium_tests_suit	Run Appiums tests suit
5	ios setup_cocoapods	Install CocoaPods
6	ios setup_portsip	Setup portsip framework for project.
7	ios generate_apn_certificate	Generate .pem and .p12
8	ios bump_project_version	Update project version number
9	ios release_internal_beta	Release internal beta to Crashlytics
10	ios release_beta	Release beta to Crashlytics
11	ios release_appstore	Deploy to App Store
12	ios code_signing	Match Certificates & Profiles
13	ios remove_keychain	
14	ios ci_build_for_testing	Build for testing
15	ios ci_unit_tests	Run Unit run_test_suite without building
16	ios ci_ui_tests	Run Native UI Testing
17	ios ci_appium_ui_tests	Run Appium UI run_test_suite without building
18	ios ci_code_signing_for_deploy	Match for deploy (CI use this lane)
19	ios ci_build_for_internal_beta	Build & achieve internal beta
20	ios ci_build_for_beta	Build & achieve external beta
21	ios ci_build_for_appstore	Build .ipa for AppStore
22	ios ci_upload_to_appstore	Upload .ipa to AppStore
23	ios ci_generate_release_note	generate release note for beta
24	ios ci_generate_release_note_for_develop_build	generate release note for develop build
25	ios ci_upload_to_crashlytics	Deploy to Crashlytics
26	ios ci_upload_dSYM	Upload dSYM to Crashlytics
27	ios ci_clean_git_repository	Discard all git changes
28	ios ci_clean_up	Clean up build artifacts, pods cache, and derived data
29	ios ci_verify_version_number	Check is Project version_number match with branch_name
30	ios ci_commit_bump_build	Commit bump build and release note
31	ios ci_push_changes	Push changes to remote origin
0	cancel	No selection, exit fastlane!

Lane's visibility for development machine

```
rphl-ios-app — bundle exec fastlane — bundle — fastlane TMPDIR=/var/folders/zg/yzx...
→ rphl-ios-app git:(chore/fastlane-sub-module) bundle exec fastlane
[✓] [15:32:53]: -----
[15:32:53]: --- Step: default_platform ---
[15:32:53]: -----
[15:32:53]: Welcome to fastlane! Here's what your app is setup to do:
+-----+
| Available lanes to run |
+-----+
| Number | Lane Name      | Description |
+-----+
| 1       | ios build_for_testing | Build test suite scheme for testing |
| 2       | ios unit_tests  | Test unit tests scheme |
| 3       | ios ui_tests    | Test ui tests scheme |
| 4       | ios tests       | Test all test target in test suite scheme |
| 0       | cancel          | No selection, exit fastlane! |
+-----+
[15:32:53]: Which number would you like run?
█
```

Lane's visibility for CI machine

A terminal window with a dark background and light text. The title bar shows 'rphl-ios-app' and some system icons. The command prompt shows the user is in a directory and has run a command to execute Fastlane. The output shows a welcome message and a table of available lanes. The user is prompted to select a lane number.

```
rphl-ios-app — IS_CI=true bundle exec fastlane — bundle — fastlane TMPDIR=/var/fold...  
→ rphl-ios-app git:(chore/fastlane-sub-module) IS_CI=true bundle exec fastlane  
[15:31:49]: -----  
[15:31:49]: --- Step: default_platform ---  
[15:31:49]: -----  
[15:31:49]: Welcome to fastlane! Here's what your app is setup to do:  
+-----+  
| Available lanes to run |  
+-----+  
| Number | Lane Name          | Description |  
+-----+  
| 1       | ios_ci_build_for_testing | Build for testing for scheme: |  
| 2       | ios_ci_run_tests      | Run test for scheme: |  
| 0       | cancel                | No selection, exit fastlane! |  
+-----+  
[15:31:49]: Which number would you like run?  
█
```



```
if !ENV.fetch('IS_CI', nil).nil?  
  import 'Parameterized.rb'  
else  
  import 'Default.rb'  
end
```

Next Steps

- Module
 - Code signing Module
 - APN Module
 - Beta distribution Module
 - App Store distribution Module
 - Utilities Module
- Unit testing for fastlane-submodule with a sample project
- Setup Continuous Integration to run unit tests for fastlane-submodule.

Thanks!

Contact Nimble

nimblehq.co

hello@nimblehq.co

Bangkok

399 Interchange 21 Sukhumvit Road, Unit
#2402-03, Klong Toei, Wattana, Bangkok
10110, Thailand

Singapore

28C Stanley St, Singapore 068737

Hong Kong

20th Floor, Central Tower
28 Queen's Road, Central, Hong Kong

