# **fastlane documentation**

# **Installation**

Make sure you have the latest version of the Xcode command line tools installed:

xcode-select --install

Install *fastlane* using

[sudo] gem install fastlane -NV

or alternatively using brew install fastlane

Before you can continue you have to modify fastlane/.env file. You have to modify # General section according.

FASTLANE\_USER=#email address of itunes#

FASTLANE\_TEAM\_NAME=#team name#

FASTLANE\_ITC\_TEAM\_NAME=#team name#

# **Available Actions**

## **Create empty app on itunes connect**

Create on Developer Portal and App Store Connect.

Before you can continue you have to modify fastlane/.env file. You have to modify # To create a new app section according.

# To create a new app

PRODUCE\_APP\_IDENTIFIER=#bundle identifier of your app#

PRODUCE\_APP\_NAME=#App name#

PRODUCE\_VERSION=#App version(ex: 1.0.0)#

PRODUCE\_SKU=#Any random string or bundle identifier#

PRODUCE\_PLATFORMS=ios

PRODUCE\_LANGUAGE=English

Then you can run below command

fastlane create\_app

**If you are doing it first time you need to ask to login in itunes connect, so you have to login in terminal so you can continue, it wont ask for second time.**

## **iOS Signing**

To generate ios distribution certificate, provising profile, sync signing certificates and provisioning profile.

Before you can continue you have to modify fastlane/.env file. You have to modify # For signin section according.

MATCH\_USERNAME=#email address of itunes#

MATCH\_GIT\_URL=#git repo url | ex: https://github.com/user/fastlaneDemo.git#

MATCH\_APP\_IDENTIFIER=#bundle identifier of your app#

MATCH\_TYPE=appstore

Then you can run below command

fastlane ios signing

## **Generate iOS APNS Certificate**

Automatically generate and renew your push notification profiles.

Before you can continue you have to modify fastlane/.env file. You have to modify # Configure Push Certificates section according.

PEM\_DEVELOPMENT=true #change to false in case of distribution

PEM\_WEBSITE\_PUSH=true

PEM\_GENERATE\_P12\_FILE=true

PEM\_FORCE=true # This will always create new certificate after this command runs.

PEM\_APP\_IDENTIFIER=#bundle identifier of your app#

PEM\_USERNAME=#email address of itunes#

PEM\_TEAM\_ID=#Team id | ex: 84JJSRPJL8#

PEM\_TEAM\_NAME=#team name#

PEM\_P12\_PASSWORD=12345678

PEM\_FILE\_NAME=apns\_live.pem

PEM\_OUTPUT\_PATH=push\_certificates

Then you can run below command to generate push after you can check push\_certificates folder, there will be three files there including pem,pkey and p12 file.

fastlane ios gen\_push\_certy

## **iOS build**

This command generates a build. You have to manually specify certificate and provisioning profile.

Before you can continue you have to modify fastlane/.env.ios file. You have to modify # Configure Code Signing and # Configure Build sections according.

# Configure Code Signing

FL\_PROJECT\_SIGNING\_PROJECT\_PATH=#Projectname.xcodeproj | ex: MyProject.xcodeproj #

FL\_PROJECT\_SIGNING\_TARGETS=#Target Name | ex: MyProject#

FL\_PROJECT\_SIGNING\_BUILD\_CONFIGURATIONS=Release

FL\_PROJECT\_USE\_AUTOMATIC\_SIGNING=false

FL\_CODE\_SIGN\_IDENTITY=Apple Distribution #You can also change to development

# Configure Build

GYM\_SCHEME=#Scheme Name | ex: MyProject#

GYM\_EXPORT\_METHOD=app-store

GYM\_OUTPUT\_DIRECTORY=build/ios #This is the output build directory

Then you can run below command to generate ipa file.

fastlane ios build

### **Release to Apple Appstore**

This will do two things, first it will build ipa file using fastlane ios build and it will upload ipa and app informations to the app store.

Before you can continue you have to modify fastlane/.env file. You have to modify # To Submit IPA to appstore section according.

DELIVER\_APP\_IDENTIFIER=#bundle identifier of your app#

DELIVER\_TEAM\_NAME=#team name#

DELIVER\_DEV\_PORTAL\_TEAM\_NAME=#team name | same as team name#

DELIVER\_RUN\_PRECHECK\_BEFORE\_SUBMIT=false

DELIVER\_SKIP\_SCREENSHOTS=false # true in case of you want to skip screenshots upload

DELIVER\_SKIP\_METADATA=false # true in case of information update will be skiped.

DELIVER\_SUBMIT\_FOR\_REVIEW=true # false in case of you want to manually submit for review.

DELIVER\_AUTOMATIC\_RELEASE=true

DELIVER\_APP\_RATING\_CONFIG\_PATH=fastlane/metadata/rating\_config.json

DELIVER\_PRICE\_TIER=0 # app price

DELIVER\_FORCE=true # skip HTML reporting

You have to add information if you want to add app metadata/information. You can find files. You have to add screenshots if you want to sync screenshots, the name of screensots should be any.

├── metadata

│ ├── copyright.txt

│ ├── en-US

│ │ ├── description.txt

│ │ ├── keywords.txt

│ │ ├── marketing\_url.txt

│ │ ├── name.txt

│ │ ├── privacy\_url.txt

│ │ ├── promotional\_text.txt

│ │ ├── release\_notes.txt

│ │ ├── subtitle.txt

│ │ └── support\_url.txt

│ ├── primary\_category.txt

│ ├── rating\_config.json

│ ├── review\_information

│ │ ├── demo\_password.txt

│ │ ├── demo\_user.txt

│ │ ├── email\_address.txt

│ │ ├── first\_name.txt

│ │ ├── last\_name.txt

│ │ ├── notes.txt

│ │ └── phone\_number.txt

│ ├── secondary\_category.txt

│ └── submission\_info.json

├── report.xml

└── screenshots

└── en-US

├── 0\_APP\_IPAD\_PRO\_3GEN\_129\_0.png

├── 0\_APP\_IPAD\_PRO\_129\_0.png

├── 0\_APP\_IPHONE\_55\_0.png

├── 0\_APP\_IPHONE\_58\_0.png

└── 0\_APP\_IPHONE\_65\_0.png

Then you have to create application specific password.

* Visit [appleid.apple.com/account/manage](http://appleid.apple.com/account/manage)
* Generate a new application specific password
* Provide the application specific password using the environment variable FASTLANE\_APPLE\_APPLICATION\_SPECIFIC\_PASSWORD in fastlane/.env.secret file.

Then you can run to release app to app store.

fastlane ios release

## **Automatically submit to beta**

This will be submit app to beta.

You can check Fastlane file under lane submit\_beta for more info.

beta\_app\_review\_info: {

contact\_email: "greg@gmail.com",

contact\_first\_name: "Greg",

contact\_last\_name: "PT",

contact\_phone: "12345678",

demo\_account\_name: "demo",

demo\_account\_password: "demo"

},

Then you can submit app using below command:

fastlane ios submit\_beta

## **Testflight**

This will build app, signin app and upload to the testflight, you have to manually add testers if any.

You can check Fastlane file under lane submit\_testflight for more info.

upload\_to\_testflight(

beta\_app\_feedback\_email: "email@email.com",

beta\_app\_description: "This is a description of my app",

demo\_account\_required: true,

notify\_external\_testers: false,

changelog: "This is my changelog of things that have changed in a log"

)

for more info please visit: <http://docs.fastlane.tools/actions/upload_to_testflight/#upload_to_testflight>