# Fastlane安装以及使用教程

Fastlane是用Ruby语言编写的一套自动化工具集和框架，每一个工具实际都对应一个Ruby脚本，用来执行某一个特定的任务，而Fastlane核心框架则允许使用者通过类似配置文件的形式，将不同的工具有机而灵活的结合在一起，从而形成一个个完整的自动化流程。

Fastlane是一套工具集，包含如下工具：

1) 测试工具

scan：自动运行测试工具，可以生成漂亮的HTML报告

2) 生成证书、配置文件工具

cert：自动创建iOS代码签名证书(.cert文件)

sigh：自动创建、更新、下载、修复Provisioning Profile

pem：自动生成、更新推送配置文件

3) 截图、描设备边框

deliver：上传截图、元数据、App到iTunesConnect

snapshot：使用UI test功能实现自动截图

frameit：在截图的图片外层套上物理设备边框

4) 自动化编译工具

gym：自动化编译工具

5) App公测工具

pilot：管理TestFlight测试用户，上传二进制文件

firim：管理firim

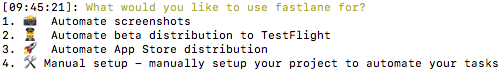
这里只介绍“自动化编译工具”其他复杂的功能不在这里讨论。

1. Fastlane安装

Fastlane是一个Ruby语言写的工具集，所以它需要Ruby环境，在安装之前需要安装Ruby。安装步骤可参考<https://www.jianshu.com/p/f7f901f5e768>。以下操作都在电脑上的终端完成。

1. 安装Ruby环境。（查看是否已经安装Ruby环境命令：ruby -v）
2. 安装最新Xcode命令行工具。（命令：xcode-select --install）
3. 执行命令：sudo gem install Fastlane 安装Fastlane。
4. Fastlane使用
5. 打开终端，cd 到工程目录（跟.xcodeproj同一个目录），执行命令：fastlane init

此时终端会提示我们选择Fastlane的什么工具



我们这里选择第4个。在终端，输入4 然后回车，然后按提示操作，直到初始化完成。此时在工程目录里会生成一个fastlane的目录，目录里面有2个文件：

../Library/Containers/com.tencent.qq/Data/Library/Application%20Support/QQ/Users/578622889/QQ/Temp.db/CBF55635-02FA-42AE-B16D-280DB1A67537.png

这2个文件都可以配置，一般情况下我们只需要配置Fastfile文件即可。

注意：如果这里选择的不是第4个，生成的fastlane目录可能不只有2个文件。有可能会生成以下6个文件，这里我们只介绍简单的打包，所以只需配置Fastfile文件。

Appfile: 存储有关开发者账号相关信息

Fastfile: 核心文件，主要用于 命令行调用和处理具体的流程，lane相对于一个方法或者函数

Deliverfile: deliver工具的配置文件

metadata: 元数据文件夹

Matchfile: Match操作对应的配置文件

screenshots: 截图文件夹

1. 配置Fastfile文件

每一个lane ~end 就是一个完整的任务。在这个配置文件里可以同时配置多个任务。

以下举例说明：

desc "打企业包" #任务描述

lane :build\_en do #任务名字，执行任务的时候用到

gym(

scheme: "Rocket.Chat", #项目名称

export\_method:"enterprise",#打包输出种类

output\_directory: "/Users/se05757/Desktop", # 打包后的 ipa 文件存放的目录

output\_name: "Rocket" # ipa文件名

)

end

上面的代码块 就是一个完整的任务：

desc 后面 是这个任务的描述，为了方便阅读，可以用中文写；

lane 后面的 build\_en 是任务的名称，后面执行fastlane build\_en这个命令就等于执行了这个任务。

export\_method 后面是方法的种类：

app-store, #AppStore正式生产环境包

ad-hoc, #生产测试包

enterprise, #企业包(299美刀账号)

development #开发测试包

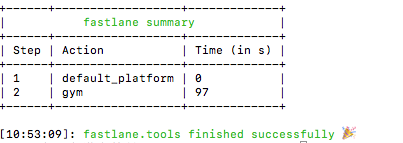
1. 执行任务的命令

fastlane build\_en

命令名称build\_en 是自定义的，是你在Fastfile配置的任务名称。

执行完命令后，就可以看到终端在编译、链接、生成ipa。

在执行命令前，要先在工程里配置好打包证书的路径，打包的版本号等信息。



下面是FastFile配置demo（仅参考）

# This file contains the fastlane.tools configuration

# You can find the documentation at https://docs.fastlane.tools

#

# For a list of all available actions, check out

#

# https://docs.fastlane.tools/actions

#

# Uncomment the line if you want fastlane to automatically update itself

# update\_fastlane

default\_platform(:ios)

platform :ios do

desc "打企业包" #任务描述

lane :build\_en do #任务名字，执行任务的时候用到

gym(

scheme: "Rocket.Chat", #项目名称

export\_method:"enterprise",#打包输出种类

output\_directory: "/Users/se05757/Desktop", # 打包后的 ipa 文件存放的目录

output\_name: "Rocket" # ipa文件名

)

end

desc "生产测试包"

lane :build\_hoc do

gym(

scheme: "Rocket.Chat", #项目名称

export\_method:"ad-hoc",#打包输出种类

output\_directory: "/Users/se05757/Desktop", # 打包后的 ipa 文件存放的目录

output\_name: "RocketHoc" # ipa文件名

)

end

desc "开发测试包"

lane :build\_dev do

gym(

scheme: "Rocket.Chat", #项目名称

export\_method:"development",#打包输出种类

output\_directory: "/Users/se05757/Desktop", # 打包后的 ipa 文件存放的目录

output\_name: "RocketDev" # ipa文件名

)

end

end