

Fastlane教程

参考文档:

[fastlane源码](#)

[fastlane actions](#)

[fastlane插件开发](#)

[使用 fastlane 实现 iOS 持续集成](#)

[深入浅出 Fastlane 一看你就懂](#)

Action是Fastlane自动化流程中的最小执行单元，直观上来讲就是Fastfile脚本中的一个命令，比如：git_pull, deliver, pod_install等等，而这些命令背后都对应一个用Ruby编写的脚本。

到目前为止Fastlane包含大约170多个Action，大约分为如下几类：

- 和移动端持续交付相关的15个核心的工具链：如：deliver（上传ipa，截屏和meta信息到ITC），supply（上传apk，截屏和meta信息到Google Play），sigh（iOS Provisioning文件管理）等等，详情如下：[fastlane Toolchain](#)
- 和iOS相关的，如：ipa, xcode_install等等
- 和Android相关的，如：gradle, adb等等
- 和版本控制相关的，如git_pull, hg_push等等
- 和iOS依赖库管理相关的，如：cocoapods, carthage等等
- 第三方平台对接相关的，如：hipchat, jira, twitter, slack等等

这些Action的详情和使用方法可以查看这个链接：[fastlane actions](#)

Plugin就是在Action的基础上做了一层包装，这个包装巧妙的利用了RubyGems这个相当成熟的Ruby库管理系统，所以其可以独立于Fastlane主仓库进行查找，安装，发布和删除。我们甚至可以简单的认为：Plugin就是RubyGem封装的Action，我们可以像管理RubyGems一样来管理Fastlane的Plugin。

- **fastlane actions:** List all available fastlane actions
- **fastlane action [action_name]:** Shows a more detailed description of an action
- **fastlane lanes:** Lists all available lanes with description
- **fastlane list:** Lists all available lanes without description
- **fastlane new_action:** Create a new action (integration) for fastlane
- **fastlane env:** Print out the fastlane ruby environment when submitting an issue

Fastlane基本操作

1 安装fastlane

```
sudo gem install fastlane
```

2 创建fastlane工程

首先进入到ios项目工程的根目录(即xcodeproj文件所在目录)，然后执行以下命令：

```
fastlane init
```

运行该命令时，会提示回答如下几个问题：

```
1.
Your Apple ID (e.g. fastlane@krausefx.com): 315701008@qq.com
App Identifier (com.krausefx.app): com.xiaoniu88.Loan

2.
Would you like to create your app on iTunes Connect and the Developer
Portal? (y/n)
n

3.
Optional: The scheme name of your app (If you don't need one, just hit Enter):

XNLoan
```

命令执行完成后，会在项目根目录下创建一个fastlane目录

3 fastlane 常用命令(以下操作都是在项目根目录下执行)

1. 查看任务列表

```
fastlane list

或

fastlane lanes
```

2. 运行某个任务，例如运行test任务

```
fastlane ios test
```

3. 查看fastlane提供的所有actions列表

```
fastlane actions
```

4. 查看指定action(例如adb)的使用帮助

```
fastlane action adb
```

5. 创建自定义action(执行如下命令, 并输入action名即可)

```
fastlane new_action
```

例如:

```
opsdeMacBook-Pro:XLLoanApp xiaoniu$ fastlane new_action
```

Must be lower case, and use a '_' between words. Do not use '.'

examples: 'testflight', 'upload_to_s3'

Name of your action: send_notify_mail

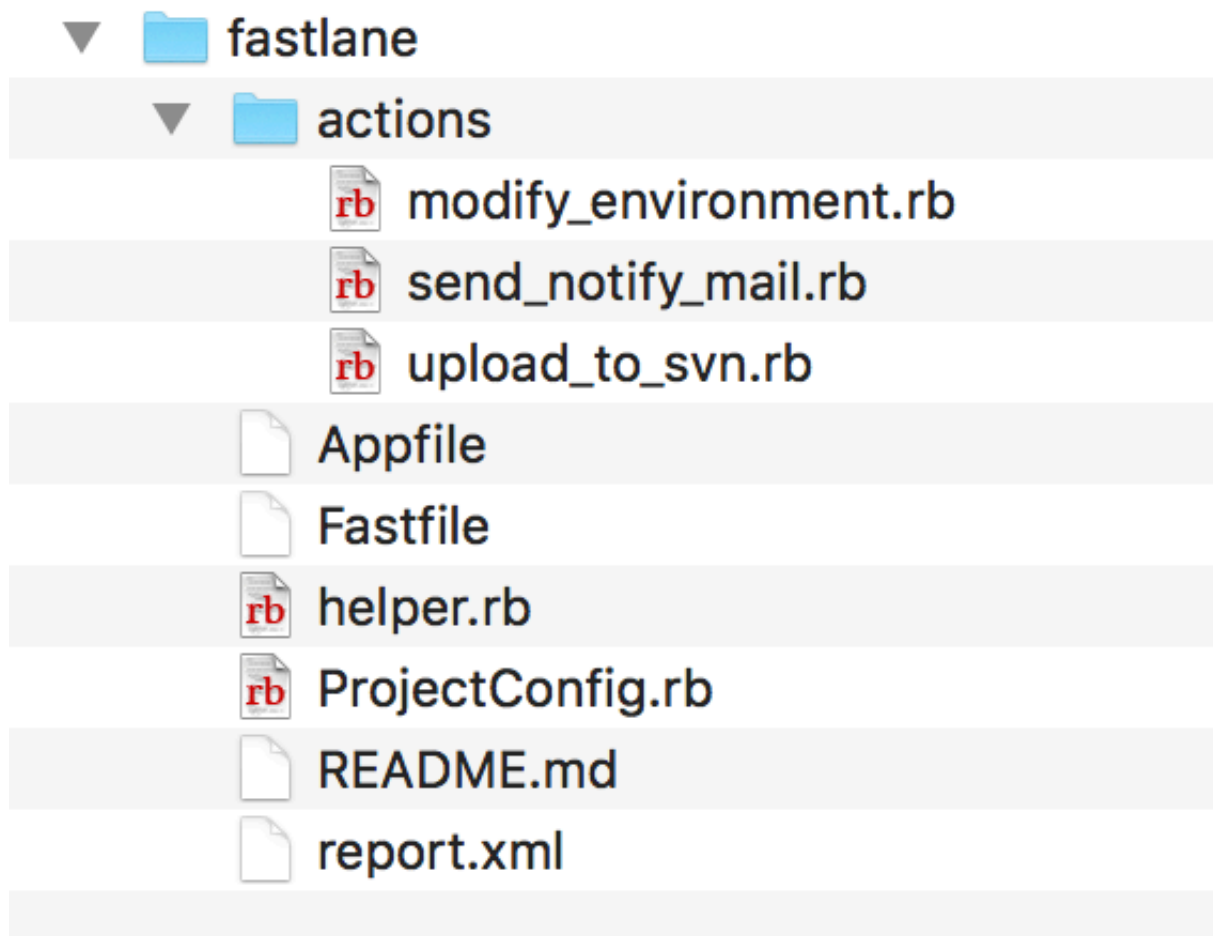
[18:02:16]: Created new action file './fastlane/actions/send_notify_mail.rb'.

Edit it to implement your custom action.

命令执行完成后, 将会在fastlane目录下创建一个actions目录, 以及以action名作为文件名的rb文件。

在线项目打包脚本

git地址: <http://git.xiaoniugit.com/zhoushicun/fastlane.git>



如上图所示, 打包脚本主要分为4部分(具体内容请看源文件):

1. 自定义action: 我们这里定义了三个自定义的action:

- modify_environment: 用于修改环境配置文件URLMacro.h
 - send_notify_mail: 用于发送打包完成的通知邮件
 - upload_to_svn: 用于将本次打出的ipa包以及DSYM文件上传到SVN
2. Fastfile文件: 所有打包流程的lane定义。
 3. helper.rb文件: 辅助函数定义, 例如解析Jenkins配置, 修改plist文件, 修改appid, 替换app图标功能(该功能需要先在App工程中内置马甲包的图标)
 4. ProjectConfig.rb文件: 所有配置项, 涵盖 环境配置, 邮箱配置, svn配置, 包渠道配置, 工程plist文件目录配置, appid, 渠道, 证书等配置。

使用教程:

选定某个分支, 将打包脚本文件夹复制到项目根目录, 并提交到git. 再配置相应的jenkins打包job即可。

fastlane开发注意:

1. 安装exchange邮件发送组件viewpoint

```
sudo gem install viewpoint

*****如果安装没有报错, 请忽略下面*****

sudo xcode-select -switch /Applications/Xcode.app/Contents/Developer

# brew install pkg-config libxml2 libxslt
brew unlink xz
gem install viewpoint
brew link xz

#nokogiri官网: https://github.com/sparklemotion/nokogiri
#安装nokogiri报错, 请参考 http://www.nokogiri.org/tutorials/installing_nokogiri.html
#gem install nokogiri
```

2. 修改plist文件, 建议使用Mac自带的 /usr/libexec/PlistBuddy 工具进行修改。因为使用plist(gem库)或者xcodesproj(gem库)在解析含有中文的plist文件时会报如下错误:(暂时还没找到解决办法)
plist-3.2.0/lib/plist/parser.rb:91:in `scan': [!] invalid byte sequence in UTF-8 (ArgumentError)