

iOS组件化之CocoaPods操作

为什么要组织此次分享？

本文档介绍了啥？

为什么要使用CocoaPods？

如何安装CocoaPods？

Podfile语法

Podspec语法

命令行相关

简介

CocoaPods是类库管理工具。它有超过2.9万个库，超过180万个app在使用。工程所需要的依赖库都在Podfile里面。CocoaPods可以解决库之间的依赖，获取源码，并把它们与Xcode工程链接到一块，去编译你的工程。

下载和安装

1. 首先安装Ruby环境

- 下载安装Xcode

它集成了Unix环境需要的开发包

- 安装RVM，命令如下：

```
$ curl -L https://get.rvm.io | bash -s stable #下载安装
$ source ~/.rvm/scripts/rvm #载入RVM环境
$ rvm -v #运行下命令，查看版本号，以检查是否安装正确
```

- 用RVM安装Ruby环境：(ruby版本需要到2.0以上，不然pod install/update不能生成xcworkspace文件)

```
$ rvm list known #列出已知的ruby版本
$ rvm install 2.2 #以2.2版本为例进行安装
```

等待一段时间，安装完成已经，可查询已经安装的ruby，命令如下：

```
$ rvm list
```

2. 修改镜像源

由于gem resources官方镜像源地址 `https://rubygems.org/` 被墙，需要修改，依次输入以下命令即可：

```
$ gem sources --remove https://rubygems.org/ #移除原有的镜像源
$ gem sources -a http://ruby.taobao.org/ #添加淘宝的镜像源
```

可验证Ruby镜像源是否替换成功，命令如下：

```
$ gem sources -l
```

正常输出如下：

```
*** CURRENT SOURCES ***

https://ruby.taobao.org
```

3.安装cocoapods

命令如下：

```
$ sudo gem install cocoapods
```

4.更新cocoapods

通常情况下，输入 `$sudo gem install cocoapods` 即可。若想安装beta版本或者未正式发布的版本，可输入以下命令：

```
$ sudo gem install cocoapods --pre
```

如果再安装不了，可尝试：

```
$ sudo gem install -n /usr/local/bin cocoapods --pre
```

5.卸载cocoapods

```
$sudo gem uninstall cocoapods
```

使用Cocoapods

1.创建Podfile

在工程所在目录下创建Podfile文件，添加依赖（首先要保证所添加的库是有效的）：

```
platform :ios, '8.0' #平台, 支持osx, ios, tvos, watchos
use_frameworks! #使用frameworks代替静态库

target 'MyApp' do #指定一个目标, 便于链接
  pod 'AFNetworking', '~> 2.7.3'
end
```

运行 `pod install`, 打开 **MyApp.xcworkspace** 即可

在工程里导入依赖, 示例如下:

```
#import <AFNetworking/AFNetworking.h>
```

```
#import "AFNetworking.h"
```

如果未使用 `use_frameworks!`, 导入依赖示例如下:

```
#import <AFNetworking/AFNetworking.h>
```

```
#import "AFNetworking.h"
```

```
#import <AFNetworking.h>
```

也可以自己指定工程, 指定工作空间, 如:

```
workspace 'Weather' #引号之间是workspace所在的路径

project 'weather/weather.xcodeproj' #cocoapods1.0以后使用, 之前使用xcodeproj

target 'weather' do
  pod 'AFNetworking', '~> 3.0'
end
```

多个目标使用相同的pod, 使用 `abstract_target`:

```
# There are no targets called "Shows" in any Xcode projects
abstract_target 'Shows' do
  pod 'ShowsKit'
  pod 'Fabric'

  # Has its own copy of ShowsKit + ShowWebAuth
  target 'ShowsiOS' do
    pod 'ShowWebAuth'
  end

  # Has its own copy of ShowsKit + ShowTVAuth
  target 'ShowsTV' do
    pod 'ShowTVAuth'
  end
end
```

由于Podfile暗含一个abstract target, 因此上面也可以简单的写成

```
pod 'ShowsKit'
pod 'Fabric'

# Has its own copy of ShowsKit + ShowWebAuth
target 'ShowsiOS' do
  pod 'ShowWebAuth'
end

# Has its own copy of ShowsKit + ShowTVAuth
target 'ShowsTV' do
  pod 'ShowTVAuth'
end
```

指定pod版本

使用最新版本:

```
pod 'AFNetworking'
```

指定某个版本:

```
pod 'AFNetworking', '2.7.3'
```

可以使用逻辑表达式, 不指定固定版本:

- '> 0.1' 大于0.1的版本
- '>= 0.1' 0.1版本和大于0.1的版本
- '< 0.1' 小于0.1的版本
- '<= 0.1' 0.1版本和小于0.1的版本

为了可以进行逻辑操作，CocoaPods有个优化的操作符 ~>:

- '~> 0.1.2' 版本0.1.2和0.2以下的版本，不包括0.2及更高的版本，相当于>=0.1.2 && < 0.2.0
- '~> 0.1' 相当于>=0.1 && < 0.2
- '~> 0' 版本0和更高版本，基于上等于没限制

指定自定义版本

使用仓库的master分支，最新版本

```
pod 'Alamofire', :git => 'https://github.com/Alamofire/Alamofire.git'
```

使用仓库的dev分支

```
pod 'Alamofire', :git => 'https://github.com/Alamofire/Alamofire.git',  
:branch => 'dev'
```

使用仓库的tag

```
pod 'Alamofire', :git => 'https://github.com/Alamofire/Alamofire.git', :tag  
=> '3.1.1'
```

或者是他用某一次提交

```
pod 'Alamofire', :git => 'https://github.com/Alamofire/Alamofire.git',  
:commit => '0f506b1c45'
```

当开发一个库的时候，使用:path，指向本地路径

```
pod 'Alamofire', :path => '~/Documents/Alamofire'
```

其它语法

1.隐藏某个库的警告

```
pod 'SSZipArchive', :inhibit_warnings => true
```

2.编译配置

```
pod 'PonyDebugger', :configurations => ['Debug', 'Beta']
```

3.不指定子仓库

```
pod 'AFNetworking'
```

会拉全部子仓库, Podfile.lock文件内容如下:

PODS:

- AFNetworking (3.1.0):
 - AFNetworking/NSURLSession (= 3.1.0)
 - AFNetworking/Reachability (= 3.1.0)
 - AFNetworking/Security (= 3.1.0)
 - AFNetworking/Serialization (= 3.1.0)
 - AFNetworking/UIKit (= 3.1.0)
- AFNetworking/NSURLSession (3.1.0):
 - AFNetworking/Reachability
 - AFNetworking/Security
 - AFNetworking/Serialization
- AFNetworking/Reachability (3.1.0)
- AFNetworking/Security (3.1.0)
- AFNetworking/Serialization (3.1.0)
- AFNetworking/UIKit (3.1.0):
 - AFNetworking/NSURLSession

DEPENDENCIES:

- AFNetworking

SPEC CHECKSUMS:

AFNetworking: 5e0e199f73d8626b11e79750991f5d173d1f8b67

PODFILE CHECKSUM: 62ce9f109426c7163a93c5cb33e94ec6900149fd

COCOAPODS: 1.2.0

4.指定子仓库，语法如下：

```
pod 'AFNetworking/Serialization'  
pod 'AFNetworking/Security'  
pod 'AFNetworking/NSURLSession'
```

或

```
pod 'AFNetworking', :subspecs => ['Serialization', 'Security',  
'NSURLSession'] #此处以Serialization, Security, NSURLSession为例
```

其它未指定的子仓库会被移除，Podfile.lock文件内容如下：

PODS:

- AFNetworking/NSURLSession (3.1.0):
 - AFNetworking/Reachability
 - AFNetworking/Security
 - AFNetworking/Serialization
- AFNetworking/Reachability (3.1.0)
- AFNetworking/Security (3.1.0)
- AFNetworking/Serialization (3.1.0)

DEPENDENCIES:

- AFNetworking/NSURLSession
- AFNetworking/Security
- AFNetworking/Serialization

SPEC CHECKSUMS:

AFNetworking: 5e0e199f73d8626b11e79750991f5d173d1f8b67

PODFILE CHECKSUM: 72d9296c101235bacf706bf9428c01cb550c46c9

COCOAPODS: 1.2.0

5.指定podspec

```
pod 'JSONKit', :podspec => 'https://example.com/JSONKit.podspec'
```

2.背后发生了啥

- 创建或更新workspace
- 把主工程添加到workspace
- 把cocoapods静态库工程添加到workspace, 如果存在就跳过
- 添加libPods.a, targets => build phases => link with libraries
- 添加cocoapods的xcode configurations 文件 (.xcconfig) 到主工程
- 改变应用的target configurations
- 向copy resources from any pods添加一个build phase, 比如script build phase
 - shell: /bin/sh
 - script: \${SRCROOT}/Pods/PodsResources.sh

3.pod install vs pod update

pod install

- 第一次安装时会创建Podfile.lock文件, 把版本锁定
- 当存在Podfile.lock时, 完全按照Podfile.lock里面列的, 下载同样的版本, 并不检测新版本
- 当从Podfile里面添加或移除pod时, 应使用pod install

pod update

- 更改pod到新版本，并更新Podfile.lock对应的版本依赖
- 用法：pod update *PODNAME* 更新某个pod库；pod update 更新所有pod库。

pod outdated

- 检测Podfile.lock中所列库对应的新版本

```
target 'MyApp' do #指定一个目标，便于链接
  pod 'AFNetworking', '~> 2.7.3' #2.7.3至2.8版本，不包括2.8版本
  pod 'FBSDKCoreKit', '~> 4.9' #4.9至5.0版本，不包括5.0版本
  pod 'Objective-C', '0.9' #固定到0.9版本??
end
```

4.Pods文件夹要纳入版本控制吗？

纳入版本控制的好处：

- 即使没有安装Cocoapods也可以编译运行，不需要pod install，不需要连网
- Pod库永远是可用的，即使远程库宕机，或消失了
- Pod库和远程仓库里是一致的

忽略Pods文件夹的好处：

- 远程仓库占用空间小
- 只要所有的pod库都可用，CocoaPods会重新创建同样的安装过程
- 即使使用不同的pod版本，不用担心冲突

无论是否控制Pods文件夹，Podfile和Podfile.lock都应该纳入版本控制。

5.Podfile.lock是啥

PODS:

- AFNetworking (3.1.0):
 - AFNetworking/NSURLSession (= 3.1.0)
 - AFNetworking/Reachability (= 3.1.0)
 - AFNetworking/Security (= 3.1.0)
 - AFNetworking/Serialization (= 3.1.0)
 - AFNetworking/UIKit (= 3.1.0)
- AFNetworking/NSURLSession (3.1.0):
 - AFNetworking/Reachability
 - AFNetworking/Security
 - AFNetworking/Serialization
- AFNetworking/Reachability (3.1.0)
- AFNetworking/Security (3.1.0)
- AFNetworking/Serialization (3.1.0)
- AFNetworking/UIKit (3.1.0):
 - AFNetworking/NSURLSession

- AKNumericFormatter (0.0.2)
- BlocksKit (2.2.3):
 - BlocksKit/All (= 2.2.3)
- BlocksKit/All (2.2.3):
 - BlocksKit/Core
 - BlocksKit/DynamicDelegate
 - BlocksKit/MessageUI
 - BlocksKit/UIKit
- BlocksKit/Core (2.2.3)
- BlocksKit/DynamicDelegate (2.2.3)
- BlocksKit/MessageUI (2.2.3):
 - BlocksKit/Core
 - BlocksKit/DynamicDelegate
- BlocksKit/UIKit (2.2.3):
 - BlocksKit/Core
 - BlocksKit/DynamicDelegate
- Colours (5.12.0):
 - Colours/ObjC (= 5.12.0)
- Colours/ObjC (5.12.0)
- DACircularProgress (2.3.1)
- DYRateView (0.0.2)
- DZNEmptyDataSet (1.7.3)
- HMSegmentedControl (1.5.3)
- IQKeyboardManager (4.0.8)
- JCore (1.1.1)
- JPush (3.0.2):
 - JCore (= 1.1.1)
- KVOController (1.2.0)
- libqrencode (3.4.2)
- Masonry (0.6.4)
- MBProgressHUD (1.0.0)
- MJExtension (3.0.13)
- MJRefresh (3.1.12)
- MWPhotoBrowser (1.0.0)
- NullSafe (1.2.2)
- OpenUDID (1.0.0)
- ReactiveCocoa (2.5):
 - ReactiveCocoa/UI (= 2.5)
- ReactiveCocoa/Core (2.5):
 - ReactiveCocoa/no-arc
- ReactiveCocoa/no-arc (2.5)
- ReactiveCocoa/UI (2.5):
 - ReactiveCocoa/Core
- SCAlipay (1.0.0):
 - SCAlipay/openssl (= 1.0.0)
 - SCAlipay/Security (= 1.0.0)
 - SCAlipay/Util (= 1.0.0)
- SCAlipay/openssl (1.0.0)
- SCAlipay/Security (1.0.0):

```
- SCALipay/Security/GTMBase64 (= 1.0.0)
- SCALipay/Security/GTMBase64 (1.0.0)
- SCALipay/Util (1.0.0)
- SCBaseFramework (1.0.2):
  - Colours (~> 5.12.0)
  - DZNEEmptyDataSet (~> 1.7.3)
  - Masonry (~> 0.6.4)
  - MJExtension (~> 3.0.9)
  - MJRefresh (~> 3.1.0)
  - ReactiveCocoa (~> 2.5)
  - SCBaseFramework/Base (= 1.0.2)
  - SCBaseFramework/Category (= 1.0.2)
  - SCBaseFramework/General (= 1.0.2)
  - SCBaseFramework/Helper (= 1.0.2)
  - SCBaseFramework/Marco (= 1.0.2)
  - SVProgressHUD (~> 2.0.3)
  - UITableView+FDTemplateLayoutCell (~> 1.4)
  - UITextView+Placeholder (~> 1.1.0)
- SCBaseFramework/Base (1.0.2):
  - Colours (~> 5.12.0)
  - DZNEEmptyDataSet (~> 1.7.3)
  - Masonry (~> 0.6.4)
  - MJExtension (~> 3.0.9)
  - MJRefresh (~> 3.1.0)
  - ReactiveCocoa (~> 2.5)
  - SVProgressHUD (~> 2.0.3)
  - UITableView+FDTemplateLayoutCell (~> 1.4)
  - UITextView+Placeholder (~> 1.1.0)
- SCBaseFramework/Category (1.0.2):
  - Colours (~> 5.12.0)
  - DZNEEmptyDataSet (~> 1.7.3)
  - Masonry (~> 0.6.4)
  - MJExtension (~> 3.0.9)
  - MJRefresh (~> 3.1.0)
  - ReactiveCocoa (~> 2.5)
  - SVProgressHUD (~> 2.0.3)
  - UITableView+FDTemplateLayoutCell (~> 1.4)
  - UITextView+Placeholder (~> 1.1.0)
- SCBaseFramework/General (1.0.2):
  - Colours (~> 5.12.0)
  - DZNEEmptyDataSet (~> 1.7.3)
  - Masonry (~> 0.6.4)
  - MJExtension (~> 3.0.9)
  - MJRefresh (~> 3.1.0)
  - ReactiveCocoa (~> 2.5)
  - SCBaseFramework/General/EmptyDataSet (= 1.0.2)
  - SVProgressHUD (~> 2.0.3)
  - UITableView+FDTemplateLayoutCell (~> 1.4)
  - UITextView+Placeholder (~> 1.1.0)
```

- SCBaseFramework/General/EmptyDataSet (1.0.2):
 - Colours (~> 5.12.0)
 - DZNEEmptyDataSet (~> 1.7.3)
 - Masonry (~> 0.6.4)
 - MJExtension (~> 3.0.9)
 - MJRefresh (~> 3.1.0)
 - ReactiveCocoa (~> 2.5)
 - SVProgressHUD (~> 2.0.3)
 - UITableView+FDTemplateLayoutCell (~> 1.4)
 - UITextView+Placeholder (~> 1.1.0)
- SCBaseFramework/Helper (1.0.2):
 - Colours (~> 5.12.0)
 - DZNEEmptyDataSet (~> 1.7.3)
 - Masonry (~> 0.6.4)
 - MJExtension (~> 3.0.9)
 - MJRefresh (~> 3.1.0)
 - ReactiveCocoa (~> 2.5)
 - SVProgressHUD (~> 2.0.3)
 - UITableView+FDTemplateLayoutCell (~> 1.4)
 - UITextView+Placeholder (~> 1.1.0)
- SCBaseFramework/Marco (1.0.2):
 - Colours (~> 5.12.0)
 - DZNEEmptyDataSet (~> 1.7.3)
 - Masonry (~> 0.6.4)
 - MJExtension (~> 3.0.9)
 - MJRefresh (~> 3.1.0)
 - ReactiveCocoa (~> 2.5)
 - SVProgressHUD (~> 2.0.3)
 - UITableView+FDTemplateLayoutCell (~> 1.4)
 - UITextView+Placeholder (~> 1.1.0)
- SCUnionPay (1.0.0)
- SDWebImage (3.7.3):
 - SDWebImage/Core (= 3.7.3)
- SDWebImage/Core (3.7.3)
- SVProgressHUD (2.0.4)
- UIAlertView-Blocks (1.0)
- UIBubbleTableView (1.0.0):
 - SDWebImage (~> 3.7.3)
- UIDeviceIdentifier (1.0.3)
- UITableView+FDTemplateLayoutCell (1.4)
- UITextView+Placeholder (1.1.1)
- UMengAnalytics (4.1.10)
- UMengUShare/Core (6.2.2):
 - UMengUShare/Network
- UMengUShare/Network (6.2.2)
- UMengUShare/Social/QQ (6.2.2):
 - UMengUShare/Core
- UMengUShare/Social/SMS (6.2.2):
 - UMengUShare/Core

- UMengUShare/Social/WeChat (6.2.2):
 - UMengUShare/Core
- VTMagic (1.2.4):
 - VTMagic/Core (= 1.2.4)
- VTMagic/Core (1.2.4)
- WebViewJavascriptBridge (6.0.2)
- YTKNetwork (2.0.3):
 - AFNetworking (~> 3.0)

DEPENDENCIES:

- AKNumericFormatter (~> 0.0.2)
- BlocksKit (= 2.2.3)
- DACircularProgress (~> 2.3.1)
- DYRateView (~> 0.0.2)
- HMSegmentedControl (~> 1.5.3)
- IQKeyboardManager (~> 4.0.7)
- JPush (~> 3.0.2)
- KVOController (~> 1.2.0)
- libqrencode (~> 3.4.2)
- Masonry (~> 0.6.4)
- MBProgressHUD (~> 1.0.0)
- MWPhotoBrowser (from `http://121.40.102.80:8888/libs/mwphotobrowser.git`, tag `1.0.1`)
- NullSafe (~> 1.2.1)
- OpenUDID (~> 1.0.0)
- Scalipay (from `http://121.40.102.80:8888/libs/scalipay.git`, tag `1.0.0`)
- SCBaseFramework (from `http://121.40.102.80:8888/hangping.lai/scbaseframework.git`)
- SCUnionPay (from `http://121.40.102.80:8888/libs/scunionpay.git`, tag `1.0.0`)
- SDWebImage (= 3.7.3)
- UIAlertView-Blocks (~> 1.0)
- UIBubbleTableView (from `http://121.40.102.80:8888/libs/uibubbletableview.git`, tag `1.0.1`)
- UIDeviceIdentifier (~> 1.0.2)
- UITableView+FDTemplateLayoutCell (~> 1.4)
- UITextView+Placeholder (~> 1.1.1)
- UMengAnalytics (~> 4.1.4)
- UMengUShare/Social/QQ
- UMengUShare/Social/SMS
- UMengUShare/Social/WeChat
- VTMagic (~> 1.2.1)
- WebViewJavascriptBridge (~> 6.0.2)
- YTKNetwork (~> 2.0.1)

EXTERNAL SOURCES:

MWPhotoBrowser:

:git: <http://121.40.102.80:8888/libs/mwphotobrowser.git>

```
:tag: 1.0.1
SCALipay:
:git: http://121.40.102.80:8888/libs/scalipay.git
:tag: 1.0.0
SCBaseFramework:
:git: http://121.40.102.80:8888/hangping.lai/scbaseframework.git
SCUnionPay:
:git: http://121.40.102.80:8888/libs/scunionpay.git
:tag: 1.0.0
UIBubbleTableView:
:git: http://121.40.102.80:8888/libs/uibubbletableview.git
:tag: 1.0.1
```

CHECKOUT OPTIONS:

```
MWPhotoBrowser:
:git: http://121.40.102.80:8888/libs/mwphotobrowser.git
:tag: 1.0.1
SCALipay:
:git: http://121.40.102.80:8888/libs/scalipay.git
:tag: 1.0.0
SCBaseFramework:
:commit: da439e8f43dfe99c070eb3a44c05386a9e9da0e3
:git: http://121.40.102.80:8888/hangping.lai/scbaseframework.git
SCUnionPay:
:git: http://121.40.102.80:8888/libs/scunionpay.git
:tag: 1.0.0
UIBubbleTableView:
:git: http://121.40.102.80:8888/libs/uibubbletableview.git
:tag: 1.0.1
```

SPEC CHECKSUMS:

```
AFNetworking: 5e0e199f73d8626b11e79750991f5d173d1f8b67
AKNumericFormatter: de969ab13c504dca7aa119b558a934ccdbd1e376
BlocksKit: db63805ebfcc217c198532a900838bffd1e400239
Colours: 4019e8eeb69d8840832eba1b3254abdbab4eec49
DACircularProgress: 4dd437c0fc3da5161cb289e07ac449493d41db71
DYRateView: 182475154ab638df0fa9080ef168bdd35f57da95
DZNEEmptyDataSet: dd7cca17c1d07dfa78956859b1e70a52505678ef
HMSegmentedControl: 7c6145afb50f26cec5f3243082f6f12b9f41318f
IQKeyboardManager: 1a8189fa153bf09044355ee0ad36826f6767e7f7
JCore: 2210a789c8efb7dc2a46b1c24d2209b995b18ad2
JPush: 5d53c88e32e48b559edc3adb20ade30903a5b9e1
KVOController: d72ace34afea42468329623b3379ab3cd1d286b6
libqrencode: 43dc34b89af83ea4b07c409cd8e09c87aa3a1c9f
Masonry: 281802d04d787ea2973179ee8bcb50500579ede2
MBProgressHUD: 4890f671c94e8a0f3cf959aa731e9de2f036d71a
MJExtension: 5932755f451458eefa24239358817f8d291240c7
MJRefresh: b96cdb21c4aa75a7b07654311ab2f315c497e806
MWPhotoBrowser: 86aaf667564cf4e46d2b028b44cc4176baf6acf8
```

```
NullSafe: cbf1f77a24558e49bc76e2673f31250d3087c792
OpenUDID: e7cb6984c7304906ad20f3a36bf087b025896b6b
ReactiveCocoa: e2db045570aa97c695e7aa97c2bcab222ae51f4a
SCAlipay: 0fcf75b21842fdef1e09b970fb2825bc2b7bfddc
SCBaseFramework: 83d31881ae7387e0a2c2734c11237942562caaec
SCUnionPay: 1cb372d3d01a2afcabc3867ed43ee1c69ba3db7c
SDWebImage: 1d2b1a1efda1adelb00b6f8498865f8ddedc8a84
SVProgressHUD: bfde85ed6c573ea802b9cb37894ed9b629034f1b
UIAlertView-Blocks: 7904744fca618afbd2ec646608924f5e69d5acd1
UIBubbleTableView: d93258d90743734cb17ac07672319dcec66bc97e
UIDeviceIdentifier: 3d1bf1348e27546c8ebeef2f02ce11ddb32730e7
UITableView+FDTemplateLayoutCell: 234e1582bcc4e18461af91155123bb96538ed030
UITextView+Placeholder: a6fc3e0378f3870036ec562a7537cbf6dbae4bf6
UMengAnalytics: 40585031b7eb085534c68a951b394731e7166257
UMengUShare: fc0a98b8d5297632e82874318fd72a10d17da228
VTMagic: b49e5f456dbcbfd9a3588ba92417233a105bc193
WebViewJavascriptBridge: 791ee0e26d1bf15efe5fb7fb9666a71a19b89d77
YTKNetwork: c4b0c29c9d639dc050d475de69ee626d53eac598
```

```
PODFILE CHECKSUM: 19a24fb4caf70d68f18a1643582ca013dee97168
```

```
COCOAPODS: 1.2.0
```

6. Pods和Submodules

git submodules和CocoaPods功能类似。submodules链接到一个特定的commit, CocoaPod捆绑到一个版本化的开发版本。

制作一个CocoaPod

创建pod

1. 创建

```
pod lib create MyLib #这里以MyLib为例
```

2. 使用tree命令查看目录树（可通过命令 `brew install tree` 安装）

```
$ tree MyLib -L 2
```

```
MyLib
├── .travis.yml #travis-ci (开源持续集成构建项目, 类似jenkins, GO) 的一个安装文件
├── _Pods.xcproject #pods工程的链接, 支持Carthage(另外一个第三方库管理工具)
├── Example #demo
│   ├── MyLib
│   ├── MyLib.xcodeproj
│   ├── MyLib.xcworkspace
│   ├── Podfile
│   ├── Podfile.lock
│   ├── Pods
│   └── Tests
├── LICENSE #默认是MIT声明
├── MyLib.podspec #库对应的Podspec
├── Pod #放库文件的地方
│   ├── Assets
│   └── Classes
│       └── RemoveMe.[swift/m]
└── README.md #markdown格式的默认readme
```

注：当向Pod/Assets或Pod/Classes文件夹添加、或移除文件、或更新podspec文件，你应当运行pod install或pod update

3.部署

部署前应该检查下Podspec文件的语言是否符合规则

```
pod lib lint #不需要连接网络
```

或

```
pod spec lint #检测外部的仓库和相关的tag
```

4.开发引入

```
pod 'Name', :path => '~/code/Pods/'
```

Podspec语法

specification规范

描述一个Pod库版本，包括去哪里获取源文件，使用什么文件，应用什么编译设置，还有一些元数据，包括名称，版本，描述等

```
Pod::Spec.new do |spec|
  spec.name           = 'Reachability' #名称
  spec.version        = '3.1.0' #版本
  spec.license         = { :type => 'BSD' } #声明
  spec.homepage        = 'https://github.com/tonymillion/Reachability' #主页
  spec.authors         = { 'Tony Million' => 'tonymillion@gmail.com' } #作者
  spec.summary         = 'ARC and GCD Compatible Reachability Class for iOS and OS X.' #pod的描述, 最多140个字
  spec.source          = { :git =>
    'https://github.com/tonymillion/Reachability.git', :tag => '3.1.0' } #检索的库的位置
  spec.source_files    = 'Reachability.{h,m}'
  spec.framework       = 'SystemConfiguration'
end
```

详细描述

name 名称, 必需

```
spec.name = 'AFNetworking'
```

version 版本, 必需

```
spec.version = '0.0.1'
```

cocoapods_version 规范支持的cocoapods版本号

```
spec.cocoapods_version = '>= 0.36'
```

author 作者, 必需

```
spec.author = 'Darth Vader'
```

```
spec.authors = 'Darth Vader', 'Wookiee'
```

```
spec.authors = { 'Darth Vader' => 'darthvader@darkside.com',
                  'Wookiee'     => 'wookiee@aggrttaaggrtt.com' }
```

license 声明, 必需

```
spec.license = 'MIT'
```

```
spec.license = { :type => 'MIT', :file => 'MIT-LICENSE.txt' }
```



```
spec.license = { :type => 'MIT', :text => <<-LICENSE
  Copyright 2012
  Permission is granted to...
  LICENSE
}
```

支持的键有, :type, :file, :text

homepage 主页, 必需

```
spec.homepage = 'http://www.example.com'
```

source 源, 必需

指定一个git源, 并指定tag

```
spec.source = { :git => 'https://github.com/AFNetworking/AFNetworking.git',
  :tag => spec.version.to_s }
```

指定以v开头的tag, 和子模块

```
spec.source = { :git => 'https://github.com/typhoon-framework/Typhoon.git',
  :tag => "v#{spec.version}", :submodules => true }
```

使用svn, 并指定tag

```
spec.source = { :svn => 'http://svn.code.sf.net/p/polyclipping/code', :tag =>
  '4.8.8' }
```

使用Mercurial(一种轻量级分布式版本控制系统), 并指定revision版本

```
spec.source = { :hg => 'https://bitbucket.org/dcutting/hyperbek', :revision
  => "#{s.version}" }
```

使用http下载一个压缩包, 支持zip, tgz, bz2, txz和tar

```
spec.source = { :http =>
  'http://dev.wechatapp.com/download/sdk/WeChat_SDK_iOS_en.zip' }
```

使用http去下载一个文件, 并使用hash值去验证下载, 支持sha1和sha256

```
spec.source = { :http =>
  'http://dev.wechatapp.com/download/sdk/WeChat_SDK_iOS_en.zip',
  :sha1 => '7e21857fe11a511f472cfd7cfa2d979bd7ab7d96' }
```

支持的键有

```
:git => :tag, :branch, :commit, :submodules
```

```
:svn => :folder, :tag, :revision
```

```
:hg => :revision
```

```
:http => :flatten, :type, :sha256, :sha1
```

```
:path
```

summary 摘要, 必需

最多140个字的pod描述

```
spec.summary = 'Computes the meaning of life.'
```

description 描述

比摘要更详细的描述

```
spec.description = <<-DESC
    Computes the meaning of life.
    Features:
    1. Is self aware
    ...
    42. Likes candies.
DESC
```

screenshots 屏幕截图

url图片列表, 用于面向UI的库, CocoaPods推荐使用gif格式

```
spec.screenshot = 'http://dl.dropbox.com/u/378729/MBProgressHUD/1.png'
```

```
spec.screenshots = [ 'http://dl.dropbox.com/u/378729/MBProgressHUD/1.png',
                    'http://dl.dropbox.com/u/378729/MBProgressHUD/2.png' ]
```

deprecated 弃用

```
spec.deprecated = true
```

platform 平台

```
spec.platform = :osx, '10.8'
```

```
spec.platform = :ios
```

deployment_target 部署目标

```
spec.ios.deployment_target = '6.0'
```

```
spec.osx.deployment_target = '10.8'
```

相对于platform键而言，它可以指定不同的部署目标

dependency 依赖

```
spec.dependency 'AFNetworking', '~> 1.0'
```

```
spec.dependency 'RestKit/CoreData', '~> 0.20.0'
```

```
spec.ios.dependency 'MBProgressHUD', '~> 0.5'
```

requires_arc 需要使用arc

默认是true

```
spec.requires_arc = false
```

```
spec.requires_arc = 'Classes/Arc'
```

```
spec.requires_arc = ['Classes/*ARC.m', 'Classes/ARC.mm']
```

framework 框架，多平台

```
spec.ios.framework = 'CFNetwork'
```

```
spec.frameworks = 'QuartzCore', 'CoreData'
```

libraries 库，多平台

```
spec.ios.library = 'xml2'
```

```
spec.libraries = 'xml2', 'z'
```

compiler_flags 编译标志位，多平台

```
spec.compiler_flags = '-DOS_OBJECT_USE_OBJC=0', '-Wno-format'
```

pod_target_xcconfig, 多平台

```
spec.pod_target_xcconfig = { 'OTHER_LDFLAGS' => '-lObjC' }
```

prefix_header_contents, 多平台

不推荐使用，会污染别的库或主工程的prefix header

```
spec.prefix_header_contents = '#import <UIKit/UIKit.h>'
```

```
spec.prefix_header_contents = '#import <UIKit/UIKit.h>', '#import  
<Foundation/Foundation.h>'
```

prefix_header_file, 多平台

不推荐使用，会污染别的库或主工程的prefix header

```
spec.prefix_header_file = 'iphone/include/prefix.pch'
```

module_name, 模块名

如不设置，默认是specification的名称

```
spec.module_name = 'Three20'
```

header_dir, 多平台

存放头文件的目录

```
spec.header_dir = 'Three20Core'
```

header_mappings_dir, 多平台

存放头文件目录的路径

```
spec.header_mappings_dir = 'src/include'
```

文件类型

模式1: * (检测文件名)

- ☐ * 匹配所有文件
- ☐ c* 匹配以c开头的文件
- ☐ *c 匹配以c结尾的文件
- ☐ *c* 匹配以包括c的文件

模式2: **

- 递归匹配目录

模式3: ?

- 匹配任何一个字符

模式4: [set]

- 匹配任何在set中的字符

模式5: {p, q}

- 匹配p, 或q

模式6: \

- 遗弃下一个元字符

```
"JSONKit.?"      #=> ["JSONKit.h", "JSONKit.m"]
"*. [a-z][a-z]"  #=> ["CHANGELOG.md", "README.md"]
"*. [^m]*"       #=> ["JSONKit.h"]
"*. {h,m}"       #=> ["JSONKit.h", "JSONKit.m"]
"*"             #=> ["CHANGELOG.md", "JSONKit.h", "JSONKit.m", "README.md"]
```

source_files 源文件, 多平台

```
spec.source_files = 'Classes/**/*.{h,m}'
```

```
spec.source_files = 'Classes/**/*.{h,m}', 'More_Classes/**/*.{h,m}'
```

public_header_files 公共头文件, 多平台

```
spec.public_header_files = 'Headers/Public/*.h'
```

private_header_files 私有头文件, 多平台

未在public headers中, 不应该暴露给主工程,

```
spec.private_header_files = 'Headers/Private/*.h'
```

vendored_frameworks, 多平台

```
spec.ios.vendored_frameworks = 'Frameworks/MyFramework.framework'
```

```
spec.vendored_frameworks = 'MyFramework.framework',
'TheirFramework.framework'
```

vendored_libraries, 多平台

```
spec.ios.vendored_library = 'Libraries/libProj4.a'
```

```
spec.vendored_libraries = 'libProj4.a', 'libJavaScriptCore.a'
```

resource_bundles, 多平台

强烈推荐使用, 可避免命名冲突, 可测试

```
spec.ios.resource_bundle = { 'MapBox' => 'MapView/Map/Resources/*.png' }
```

```
spec.resource_bundles = {  
  'MapBox' => ['MapView/Map/Resources/*.png'],  
  'OtherResources' => ['MapView/Map/OtherResources/*.png']  
}
```

resources 资源文件, 多平台

推荐使用resource bundles

```
spec.resource = 'Resources/HockeySDK.bundle'
```

```
spec.resources = ['Images/*.png', 'Sounds/*']
```

exclude_files, 多平台

```
spec.ios.exclude_files = 'Classes/osx'
```

```
spec.exclude_files = 'Classes/**/unused.{h,m}'
```

preserve_paths, 多平台

下载后不被移除的文件, 默认的是, cocoapods会移除其它类型的文件

```
spec.preserve_path = 'IMPORTANT.txt'
```

```
spec.preserve_paths = 'Frameworks/*.framework'
```

module_map, 多平台???

```
spec.module_map = 'source/module.modulemap'
```

子库

subspec

```
subspec 'Twitter' do |sp|  
  sp.source_files = 'Classes/Twitter'  
end  
  
subspec 'Pinboard' do |sp|  
  sp.source_files = 'Classes/Pinboard'  
end
```

使用方法

```
pod 'ShareKit/Twitter', '2.0'  
pod 'ShareKit/Pinboard', '2.0'
```

default_subspecs

```
spec.default_subspec = 'Core'
```

```
spec.default_subspecs = 'Core', 'UI'
```

命令行相关

安装

```
$ touch Podfile  
$ vi Podfile  
$ pod install
```

```
$ open *.xcworkspace
```

pod init

生成默认的podfile文件

```
pod init XCODEPROJ #xcodeproj为工程名, project会在podfile里面指定
```

举例: pod init weather.xcodeproj

pod install

下载podfile中所有的依赖, 并在./Pods中创建一个xcode pods library工程

选项:

—repo-update 在安装前更新pod仓库, 先运行pod repo update

—no-repo-update 在安装前不更新pod仓库

—project-directory=/project/dir/ 工程根目录路径

pod update

```
pod update [POD_NAMES ...] #如果不指定pod_names, 将更新所有pod
```

使用: pod update AFNetworking CommontTools

选项:

—repo-update 在安装前更新pod仓库, 先运行pod repo update

—no-repo-update 在安装前不更新pod仓库

—project-directory=/project/dir/ 工程根目录路径

pod outdated

```
pod outdated
```

列举podfile.lock中过时的pod, 只检测spec仓库, 不包括本地或外地的源

选项:

—repo-update 在安装前更新pod仓库, 先运行pod repo update

—no-repo-update 在安装前不更新pod仓库

—project-directory=/project/dir/ 工程根目录路径

pod deintegrate

v1.0.0.beta.1之后可用

```
pod deintegrate [XCODE_PROJECT] #如果不指定, 将会从当前目录寻找xcode工程
```

从cocoapods分离主工程, 从主工程移除所有cocoapods的痕迹

选项:

—project-directory=/project/dir/ 工程根目录路径

pod env

显示pod环境

pod search

v0.0.2以上可用

```
pod search QUERY #查找pod
```


query为要查询的内容，不分大小写，将会查找name, summary, description或作者。

选项：

—regex 把query当作正则表达式

—simple 只按名称搜索pod库

—ios/osx/watchos/tvos 搜索适用某个平台的库

—no-pager 搜索结果不要分页

—web 在cocopods.org上搜索

—显示额外的统计，如github上的watchers和forks

pod list

列出所有可用的pod

```
pod list
```

选项

—update 在列举前进行更新，即运行pod repo update

—显示额外的统计，如github上的watchers和forks

pod try

v0.29.0之后可用

```
pod try NAME|URL
```

按pod名或pod的git url下载，安装依赖，并打开demo工程

选项：

—podspec_name=[name] 指定podspec文件的名称，如提供url地址，可在后面跟上该选项

—no-repo-update 在安装前不更新pod仓库

pod spec create

```
pod spec create [NAME|https://github.com/USER/REPO]
```

在当前文件夹下创建一个podspec，如果github url是可用的话，将会预填一部分内容到spec文件中，

如：pod spec create <https://github.com/hengyizhangcn/EditView>

pod spec lint

```
pod spec lint [NAME.podspec|DIRECTORY|http://PATH/NAME.podspec ...]
```

验证podspec文件，参数可以不提供，验证当前文件夹

选项：

—quick 跳过需要下载和编译的步骤

—allow-warnings 允许警告

—subspec=NAME 只验证给定的subspec

—no-subspecs 路过验证subspec

—no-clean 将构建目录保留完好以供检查

—fail-fast 在第一个平台或subspec失败时即停止

—use-libraries 使用静态库安装spec

—private 只检测私有spec

—swift-version=VERSION 指定swift版本

--sources=<https://github.com/artsy/Specs>,master 下拉的依赖库中的源（默认是<https://github.com/CocoaPods/Specs.git>）。多个源之间用逗号分隔

pod spec cat

```
pod spec cat [QUERY]
```

打印名称匹配query的podspec内容

选项：

—regex 把query当作正则表达式

—show-all 选自给定podspec的所有版本

pod spec which

```
pod spec which [QUERY]
```

打印名称匹配query的podspec路径

选项：

—regex 把query当作正则表达式

—show-all 选自给定podspec的所有版本

pod spec edit

```
pod spec edit [QUERY]
```

打开名称匹配query的podspec以供编辑

选项:

—regex 把query当作正则表达式

—show-all 选自给定podspec的所有版本

pod lib create

```
pod lib create NAME
```

选项:

—template-url=URL git repo的URL包含一个兼容性的模板

pod lib lint

验证库使用的文件

—quick 跳过需要下载和编译的步骤

—allow-warnings 允许警告

—subspec=NAME 只验证给定的subspec

—no-subspecs 路过验证subspec

—no-clean 将构建目录保留完好以供检查

—fail-fast 在第一个平台或subspec失败时即停止

—use-libraries 使用静态库安装spec

—private 只检测私有spec

—swift-version=VERSION 指定swift版本

--sources=<https://github.com/artsy/Specs>,master 下拉的依赖库中的源（默认是<https://github.com/CocoaPods/Specs.git>）。多个源之间用逗号分隔

pod cache list

```
pod cache list [NAME] #name可不提供
```

显示pods缓存的内容，以YAML树的形式输出

选项:

—short 只打印相对路径

pod cache clean

```
pod cache clean [NAME]
```

选项:

—all 清除所有缓存的pods

注：要么指定一个pod名，要么带上—all参数

Trunk

Repos

IPC

Plugins

经常碰到的问题

1.连接超时

```
[!] /usr/bin/git clone https://github.com/jpush/jpush-ios-sdk-pod.git
/var/folders/8w/j6x5f1290zq_g0q8b0dp7fzm0000gn/T/d20170227-15744-1kyd22x --
template= --single-branch --depth 1 --branch 2.2.0.1
Cloning into '/var/folders/8w/j6x5f1290zq_g0q8b0dp7fzm0000gn/T/d20170227-
15744-1kyd22x'...
error: RPC failed; curl 56 SSLRead() return error -36
fatal: The remote end hung up unexpectedly
fatal: early EOF
fatal: unpack-objects failed
```

2.找不到pod

```
Unable to find a pod with name, author, summary, or description matching.
```

删除索引即可:rm ~/Library/Caches/CocoaPods/search_index.json

3.当存在多个Xcode版本

```
[!] Unable to add a source with url `https://github.com/CocoaPods/Specs.git`
named `master-1`.
You can try adding it manually in `~/.cocoapods/repos` or via `pod repo add`.
```

在Xcode->Preference->locations里面选择command tools line

4.target未添加

```
The dependency is not used in any concrete target.
```

当pod升级后需要在pod file添加target，如target 'MyProject' do *** end

在添加target或者删除target时, 需要先运行pod deintegrate, 不然Build Phases中的Embed Pods Frameworks的shell路径会改变

参考

<https://github.com/CocoaPods/CocoaPods/wiki>

<http://guides.cocoapods.org/using/pod-install-vs-update.html#introduction>

如何使用Carthage管理iOS依赖库

<http://www.jianshu.com/p/5ccde5f22a17>

<http://guides.cocoapods.org/making/specs-and-specs-repo.html>

http://guides.cocoapods.org/syntax/podspec.html#header_mappings_dir

<http://guides.cocoapods.org/making/private-cocoapods.html>

http://guides.cocoapods.org/syntax/podspec.html#group_multi_platform_support

<http://guides.cocoapods.org/terminal/commands.html#commands>

萤石开放平台iOS SDK接口使用说明

<https://open.ys7.com/doc/book/4.x/ios-sdk.html>

鹰眼监控gitlab远程库

<http://121.40.102.80:8888/libs/eaglemonitoring>

Edit by 张恒一

Mar 24, 2017