

# CocoaPods 使用总结(小白版本)

by: Magic\_Unique 冷秋

SinaWeibo: [@Magic-Unique](#)

Github: [Magic-Unique](#)

特别鸣谢: bo同学, 提供第五节的内容.

## 前言

---

小石头在1509班浪的一逼, 结果群里好多同学都是稍微一不留神就跟不上了, 一整天都是各种 pod 安装失败.

所以我写了一篇比较通俗易懂的文章, 来多方面介绍 CocoaPods 这个工具.

如果你觉得篇幅较长, 你可以着重看前面带有竖线的部分的内容.

## 目录

---

- 简介
  - Github
  - 第三方框架
  - CocoaPods
- CocoaPods 安装
  - 为你的 Mac 安装 CocoaPods
  - gem ruby 配置
  - CocoaPods 安装
- CocoaPods 导入
  - 什么样的项目需要 CocoaPods
  - 在项目中第一次使用 CocoaPods
- CocoaPods 修改
  - 何时需要修改 CocoaPods
  - 修改 CocoaPods 的配置
- CocoaPods 无视警告

## 简介

---

## Github

Github 是一个基于 git 的类似社区的代码管理平台. 在这个平台上, 任何语言(包括任何系统)的开发者可以将他的代码上传至此. Github 会利用 git 工具帮开发者管理好代码, 相当于是一个云盘. 这些上传的代码是公开的, 任何人(不只是上传代码的开发者)都可以下载任何开发者上传的代码. 也因此, 很多牛逼的人喜欢把自己写好的一些工具类放在上面, 以供别人使用和维护. 同样, 一些初学者也可以来此下载大神们的代码来学习.

Github 已经成为全球最大的开发者平台, 希望你能多多使用.

(如果你不喜欢自己在 Github 上的代码被公开, 你可能需要支付给 Github 一定费用)

(git 版本管理工具, 如果你不知道他是什么, 可以百度一下, git和svn会在你未来的开发工作中使用到, 学习他们是必不可少的)

## 第三方框架

在 iOS 开发中, 你会发现一些牛逼的人物经过长时间的维护和探索, 只做了了一些好多工具类. 这些工具可以在你很多项目中发挥着巨大的作用. 而你自己如果也去写一样的工具, 则需要花很多时间. 好在那些工具类都是在 Github 上开源的, 你可以下载来自己使用. 我们称这些工具类叫做**第三方框架**

## CocoaPods

CocoaPods 是一个专门用于 Xcode 项目中, 管理那些来自 Github 上的第三方框架.

你会问:"我同样从 Github 上 `Download zip` 后, 提取框架核心代码, 直接拉到 Xcode 工程中不行吗?"

答案是可以的, 但是如果出现以下情况, 你就只能 20 投了.

### 框架使用框架

好吧我不知道应该取什么标题, 你们将就看一下. 大意就是一个框架用到了另一个框架, 则你需要同时拥有两个框架的源文件.

举个栗子:

小石头写了一个框架叫 `MeiLingFramework`.

之后大白觉得这个框架很不错, 不过似乎可以将这个框架改造一下做一些更牛逼的功能. 于是, 大白基于这个框架, 写了另一个框架叫 `XiaoBaiFramework`.

后来, 小猫发现了 `XiaoBaiFramework`, 感觉不错, 又脑洞大开想利用这个框架再开发一个框架. 于是 `MaleFramework` 诞生了.

现在你想使用 `MaleFramework`, 于是把 `MaleFramework` 的所有源文件都导入到你的工程中, 发现他报错了. 原因是 "XiaoBaiFramework" not found, 找不到这个框架. 于是你找到小猫写的使用说明, 发现要用 `MaleFramework` 的时候必须要有 `XiaoBaiFramework`. 于是你去下载

了 `XiaoBaiFramework` 导入到工程中. 但是又报错了, 这次的原因是 "MeiLingFramework" not found. 没办法, 你只好查阅了小白的使用说明, 然后导入了小石头写的 `MeiLingFramework` .

世界终于太平了.

如果出现这种情况怎么办呢? 在没有 CocoaPods 的情况下, 你只能按照上面的例子来做. 但是现在有了 CocoaPods, 你只要导入 `MaleFramework` , CocoaPods 会自动帮你导入他需要的框架, 以及那些框架需要的别的框架.

## 框架更新

既然有人维护, 那么就会发现这些工具类的 bug, 一旦 bug 出现, 就必须解决掉, 因为全球有很多人使用他, 必须要保证他的安全性.

于是, 第三方框架的更新就是必不可少的, 说不定某个框架在你使用时候就更新了呢, 而你还在使用当初下载时候的旧版本.

这种情况你只能喊爹了. 而且喊了爹也没用.

现在有了 CocoaPods, 他会帮你更新你使用的所有第三方框架, 只是一个终端命令, 全部都搞定.

# CocoaPods 的安装

---

## 为你的 Mac 安装 CocoaPods

CocoaPods 是一个软件, 或者说是一个工具. 它是以一个可执行文件存在你的 Mac 电脑上, 就像你 Mac 上的 vim 编辑器一样.

不过 Mac 是默认不安装 CocoaPods 的, 所以要让你的电脑拥有这个可执行文件, 就必须安装 .

CocoaPods 是用 ruby 写的程序, 被分成了诺干的 gem 包(这两个词你不懂就算了).

安装 CocoaPods 一般通过 Mac 自带的 ruby 环境安装, 也就是 gem. 但是 gem 被天朝的那堵墙给挡住了. 所以我们要先改造 gem, 让他拥有一个 "翻墙能力".

## gem ruby 配置

其实要让 gem 拥有翻墙的能力很简单, 一个 VPN 就能搞定. 但是并不是所有人都有 VPN.

淘宝为 gem 做了一个镜像. 淘宝每过几分钟, 通过他的手段, 将 gem 的另一端数据完全拷贝到淘宝的服务器上, 这样淘宝就模拟了一个镜像, 而这个镜像中, 就有 CocoaPods 的可执行程序.

我们要做的是, 直接去淘宝镜像下载 CocoaPods.

所以我们要告诉 gem, "你丫的别用你自己的镜像了, 用淘宝的!"

打开 终端

输入 `gem source`

我们要做的第一件事, 就是告诉他别用自己的镜像.

在终端中输入

```
gem sources -l
```

(注意, 最后一个是小写的L)

然后你就会发现, 终端输出

```
*** CURRENT SOURCES ***  
  
https://rubygems.org/
```

说明此时 `gem` 使用的是自己的镜像

继续终端输入

```
gem sources --remove https://rubygems.org/
```

将这个原有的镜像删除, 然后接下来继续输入

```
gem sources -a https://ruby.taobao.org/
```

添加淘宝镜像, 最后再次输入

```
gem sources -l
```

查看淘宝镜像是否成功加入

我们成功的配置好 `gem ruby` 环境了, 这样他就有"伪翻墙能力"

## CocoaPods 安装

在成功"翻墙"以后, 就可以从淘宝安装 `CocoaPods` 了

在终端中输入

```
sudo gem install cocoapods
```

然后提示 `Password:`, 此时输入你的 Mac 密码(不会显示你的输入结果), 然后回车.

当你当前 Mac 的用户名出现在终端最后一行, 也就是你又可以继续输入命令的时候, CocoaPods 就安装完毕了.

## CocoaPods 导入

在你已有 CocoaPods 了, 你就可以去使用 CocoaPods 的几个命令.

在这里先介绍几个常用的命令, 并在之后的篇幅中详细介绍他们的使用

```
pod init      //为一个项目创建 Podfile 文件
pod install   //为一个项目创建 Pod 环境
pod update    //为一个项目更新 Pod 环境
```

## 什么样的项目需要 CocoaPods

其实很简单, 需要方便管理第三方框架的项目就需要 CocoaPods.

## 在项目中第一次使用 CocoaPods

打开终端

通过 cd 命令进入到你的工程根目录, 也就是 Xcode 生成的 .xcodeproj 文件所在目录

然后在终端输入

```
pod init
```

此时 CocoaPods 会自动为你的工程创建一个 Podfile 文件

之后你可以编辑这个 Podfile 文件. 对于第一次编辑, 我比较喜欢用 vim 编辑器, 这样可以防止中文符号.

使用 vim 编辑器编辑 Podfile 文件, 在终端输入

```
vi Podfile
```

你就可以在终端上看到 Podfile 文件里面的内容了.

然后你需要按一下 **i** 键, 进入编辑模式

将光标移动到 do 与 end 之间的空行中, 输入

```
pod 'AFNetworking'
pod 'MJExtension'
```

这样做的目的是, 告诉 CocoaPods, 我要用到 `AFNetworking` 和 `MJExtension` 框架, 同理, 如果你现在需要别的框架, 也按这格式写.

如果之后需要 **添加/删除/更新** 框架, 请看下一节.

然后按一下键盘上的 `esc` 键, 再按 `shift + ;`, 输入 `wq` 保存并退出.

我们告诉了 CocoaPods 我们需要的框架, 接下来就是这个项目第一次使用 CocoaPods, 所以要为这个项目搭建 CocoaPods 环境.

在终端输入

```
pod install
```

CocoaPods 就会读取刚才的 `Podfile` 文件, 并且去 Github 上搜索里面的框架, 然后全部配置好, 这个过程可能很长, 请耐心等待.

之后, 如果你看到了啤酒的符号, 说明你的 CocoaPods 环境搭建成功.

此时, 在你的项目中会多出很多文件夹, 以及一个很重要的文件 `.xcworkspace`.

现在, 你就不能使用 **蓝色** 的 `.xcodeproj` 来打开项目, 而都要使用 **白色** 的 `.xcworkspace` 文件来打开项目

配置好 CocoaPods 环境后, 你就可以重新打开项目了, 记得要用白色的工程文件打开, 因为它才包含有 CocoaPods 的配置信息.

如果你打开后发现报错, 原因是第三方框架找不到, 请检查你是不是使用蓝色的工程文件打开项目.

在这里要注意的是

如果你是用 CocoaPods 导入框架的, 在代码中引入框架头文件请用 `#import <>`.

如果你是直接拖代码到项目中的, 请用 `#import ""`

## CocoaPods 修改

CocoaPods 环境在第一次使用的时候已经搭建好了, 但是我们在搭建环境的时候肯定不知道未来开发会出现什么情况, 所以难免会去修改环境.

### 何时需要修改 CocoaPods

1. 添加框架
2. 删除框架
3. 更新框架

## 修改 CocoaPods 的配置

第二次使用 CocoaPods 的时候就不需要 vim 编辑器了, 因为我们在 Xcode 中可以直接看到 Podfile 文件.

打开 Xcode 左边栏的文件导航栏, 展开第二个蓝色块, 也就是 Pods 标题的那个.

你会看到 Podfile 文件

点击他, 在 Xcode 窗口中间的编辑器中, 按照格式编辑里面的框架列表.

你可以删除框架, 也可以另起一行添加框架, 如果你要更新框架就可以直接下一步.

编辑好后, 打开终端, cd 到你的项目根目录

输入

```
pod update
```

CocoaPods 就会帮你更新你的框架列表, 删除不用的框架, 下载需要的框架, 同时更新已有的框架.

并会在终端中输出结果, 绿色的代表有更新(或者添加), 黑色代表没有更新.

## CocoaPods 无视警告

最近有些框架没有更新好, 或者说进入了一些苹果API的淘汰缓冲期, 比如 `AFNetworking` 的 `NSURLOperation`, 所以导致 Xcode 会对第三方框架报警告, 对于处女座的孩子们, 如果不想看到这些警告也是可以的.

编辑 Podfile 文件 `do` 所在的上面一行填写

```
inhibit_all_warnings!
```

然后到终端执行一次 `pod update` 即可.