




CocoaPods私有库创建方法总结


 CocoaPods私有库创建官方教程：<https://guides.cocoapods.org/making/private-cocoapods.html>

 注意：创建CocoaPods私有库和源码库统一在Gitlab的MOJi CocoaPods下创建。

 创建CocoaPods私有库有一个全局了解，主要步骤总结如下：

1. 创建私有Specs仓库
2. 创建源码库，并git clone到本地
3. 创建CocoaPods私有库模板
4. 编写CocoaPods私有库podspec文件
5. 添加源码和资源文件到指定的文件夹中（每次更改都需要在Example文件夹中pod install一下）
6. 验证podspec文件是否无误
7. 移动CocoaPods私有库模板中的文件到源码库
8. 提交源码到git仓库，并打tag
9. 将 podspec 提交到spec仓库
10. 在项目中使用私有库

1、创建私有Specs仓库

 Specs创建一次即可，后期所有项目的cocoapod.spec文件都会上传到这个仓库



Specs

Project ID: 188



0



0

2 Commits 1 Branch 0 Tags 184 KB Files 184 KB Storage



Auto DevOps

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.

Learn more in the [Auto DevOps documentation](#)

Enable in settings

master

Specs /



History

Find file

Web IDE



Clone



[Add] MOJiBase (1.3.11)

zhaohanjun authored 1 day ago

93f0745d



README

Add LICENSE

Add CHANGELOG

Add CONTRIBUTING

Add Kubernetes cluster

Set up CI/CD

执行repo 命令添加私有库Repo 打开终端，在任意目录下执行下面的命令：

Dockerfile

```
1 pod repo add Specs http://192.168.1.3/moji-cocoapods/Specs.git
2
3 解释：
4 pod repo add 【私有库名称】 【1.1中创建的远程仓库的git地址】
```

如果需要更改，用以下命令删除并重复上一步

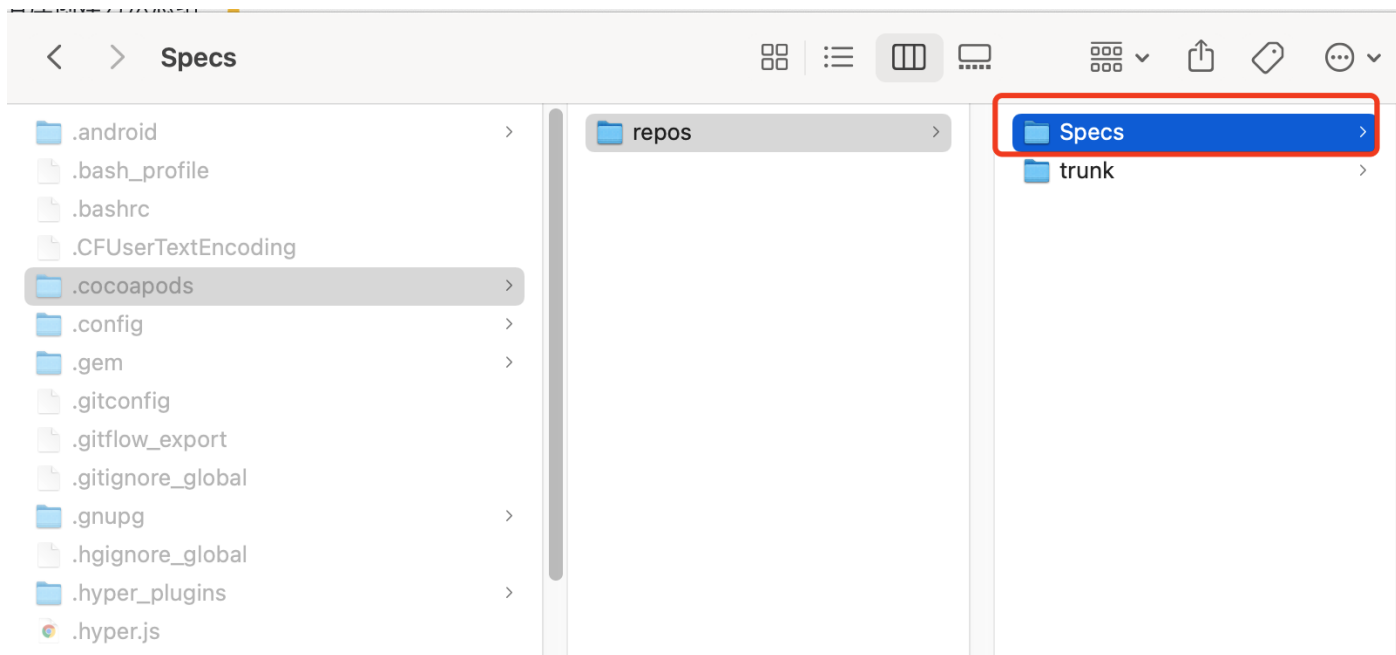
C#

```
1 pod repo remove REPO_NAME
```

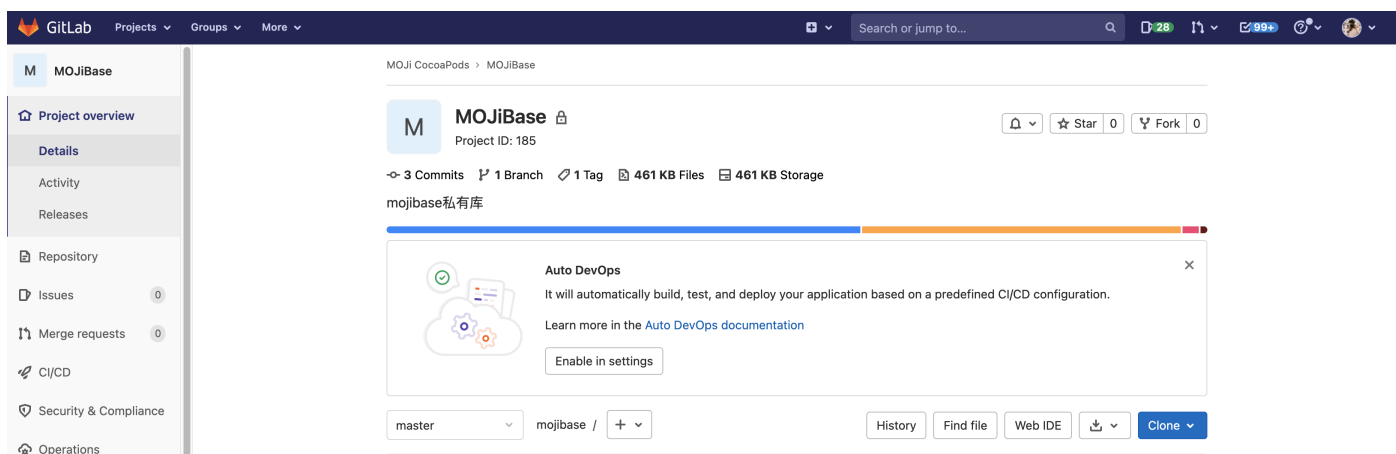
成功后，我们将在以下目录中看到我们所创建的私有库文件夹

JavaScript

```
1 ~/.cocoapods/repos
```

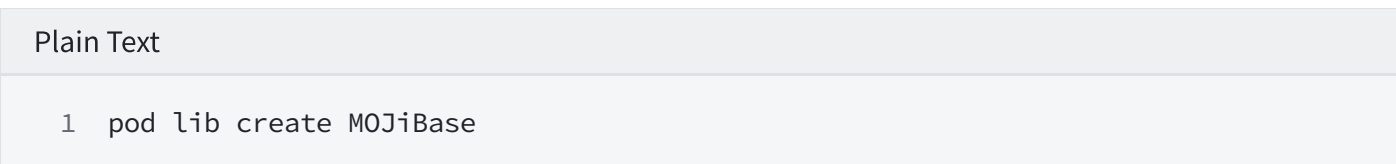


2、在MOJi CocoaPods下创建源码库，并git clone到本地



3、创建Cocoapods私有库模板

cd到你要储存项目的文件夹，使用以下命令创建Cocoapods私有库模板



紧接着，会有一些参数需要配置：（语言根据实际情况选择，生成的结构都是一样的，只是生成的默认语言会不一样）

```
-----
To get you started we need to ask a few questions, this should only take a minute.

2019-08-30 17:21:31.054 defaults[7653:540420]
The domain/default pair of (org.cocoapods.pod-template, HasRunBefore) does not exist
If this is your first time we recommend running through with the guide:
- https://guides.cocoapods.org/making/using-pod-lib-create.html
( hold cmd and click links to open in a browser. )

Press return to continue.

What platform do you want to use?? [ iOS / macOS ]
> ios

What language do you want to use?? [ Swift / ObjC ]
> swift

Would you like to include a demo application with your library? [ Yes / No ]
> yes

Which testing frameworks will you use? [ Quick / None ]
> none

Would you like to do view based testing? [ Yes / No ]
> no

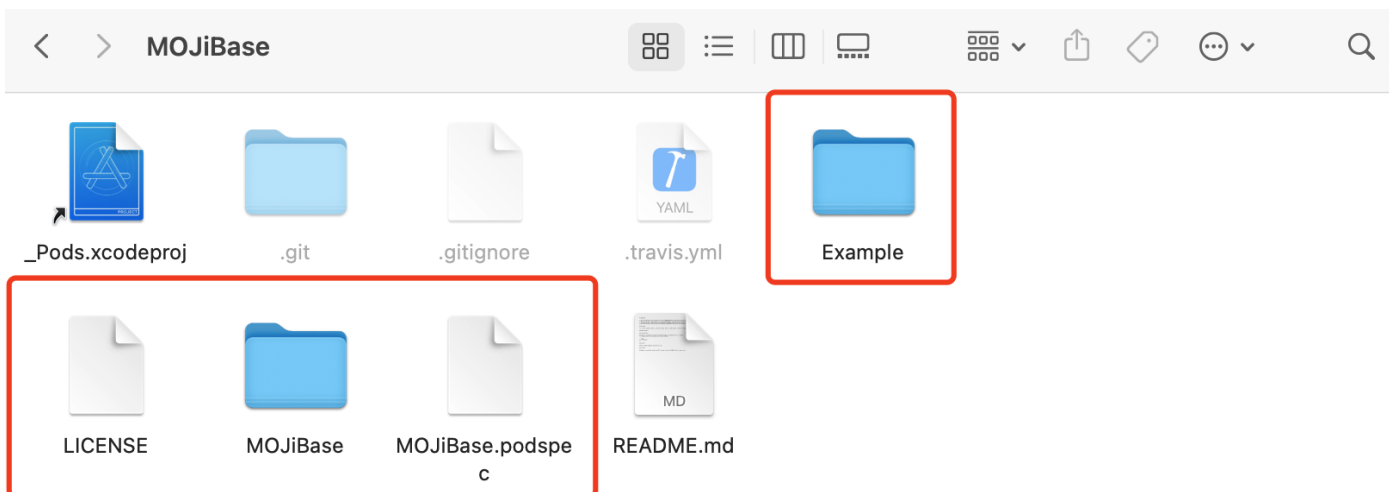
Running pod install on your new library.
```

选择平台
选择项目语言，这里输入swift
选择是否包含一个demo应用。一般选择yes，便于测试自己的组件是否可用
使用什么测试框架，这里选择none
是否要进行基于视图的测试，这里选择no

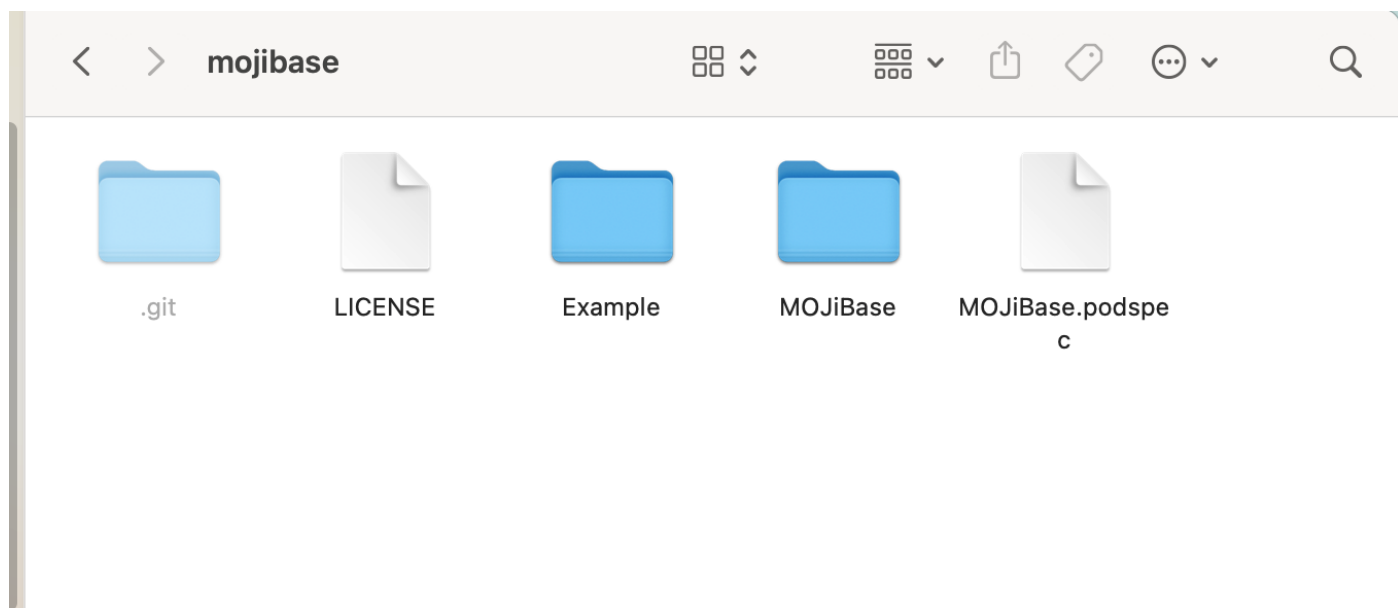
@掘金技术社区

配置完成后，会自动打开创建的项目

4、将pod创建的项目文件copy到MOJiBase本地仓库里



最终如如下图所示：



5、配置podspec文件

打开 MOJiBase本地仓库中的 Example 中的工程

选择 MOJiBasepodspec 文件进行编辑：

 相关字段可以查询官方文档【[Podspec Syntax Reference](#)】。

以下是MOJiBase.podspec文件示例，仅供参考。

Bash

```
1 #
2 # Be sure to run `pod lib lint MOJiBase.podspec` to ensure this is a
3 # valid spec before submitting.
4 #
5 # Any lines starting with a # are optional, but their use is encouraged
6 # To learn more about a Podspec see https://guides.cocoapods.org/syntax/podspec.html
7 #
8
9 Pod::Spec.new do |s|
10   s.name           = 'MOJiBase'
11   s.version        = '1.3.11'
12   s.summary        = 'MOJi基础工具类库，其中包括一些UI功能，后期需要剥离'
13   s.swift_version  = '5.0'
14
15   # This description is used to generate tags and improve search results.
16   # * Think: What does it do? Why did you write it? What is the focus?
17   # * Try to keep it short, snappy and to the point.
18   # * Write the description between the DESC delimiters below.
19   # * Finally, don't worry about the indent. CocoaPods strips it!
```

```

20
21 s.description = <<-DESC
22 TODO: Add long description of the pod here.
23     DESC
24
25 s.homepage = 'http://192.168.1.3/moji-cocoapods/mojibase.git'
26 # s.screenshots = 'www.example.com/screenshots_1', 'www.example.com/screenshots_2'
27 s.license = { :type => 'MIT', :file => 'LICENSE' }
28 s.author = { '赵汉军' => 'zhaohanjun@mail.shareintelli.com' }
29 s.source = { :git => 'http://192.168.1.3/moji-cocoapods/mojibase.git',
30             :tag => s.version.to_s }
31 # s.social_media_url = 'https://twitter.com/<TWITTER_USERNAME>'
32
33 s.ios.deployment_target = '11.0'
34
35 s.source_files = 'MOJiBase/Classes/**/*'
36
37 s.resource_bundles = {
38   #'MOJiBase' => ['MOJiBase/Assets/*.png']
39   #'MOJiBase' => ['MOJiBase/base-images/*.png']
40   #'MOJiBase' => ['MOJiBase/**/*.{xib,bundle}']
41   #'MOJiBase' => ['MOJiBase/Assets/base-images.bundle']
42 }
43 # s.public_header_files = 'Pod/Classes/**/*.*h'
44 s.frameworks = 'UIKit', 'Foundation'
45 s.dependency 'SVProgressHUD'
46 s.dependency 'SwiftyJSON'
47 s.dependency 'SnapKit'
48
49 end

```

相关字段解释：

Bash

```
1  #私有库对应的版本号
2  s.version      = '1.3.11'
3  #私有库描述, 最好写一下, 不然最终验证的时候会报警告⚠
4  s.summary      = 'MOJi基础工具类库, 其中包括一些UI功能, 后期需要剥离'
5  #如果使用的是swift, 需要指定使用的版本, 否则有可能编译不过, 最终验证也会报警告
6  s.swift_version = '5.0'
7  #仓库地址, 必须得写, 否则后面spec文件会上传不成功
8  s.source       = { :git => 'http://192.168.1.3/moji-cocoapods/mojibase.git', :t
  ag => s.version.to_s }
9  #源码文件路径
10 s.source_files = 'MOJiBase/Classes/**/*'
11 #指定的资源文件读取路径, xib、png、bundle、Assets都需要指定
12 s.resource_bundles = {
13   'MOJiBase' => ['MOJiBase/**/*.{xib,bundle}']
14 }
15 #依赖的系统库
16 s.frameworks = 'UIKit', 'Foundation'
17 #依赖的第三方库
18 s.dependency 'SnapKit'
19
```

如果依赖的库是某个库的静态库, 也就是framework, 则需要在.podspec文件中的添加依赖前进行指明, 这种情况是我在使用高德地图相关功能时遇到的, 地图包里面都是framework, 如下:

Bash

```
1  s.static_framework = true
2
3  s.dependency 'AMap3DMap-NO-IDFA'
4  s.dependency 'AMapFoundation-NO-IDFA'
5  s.dependency 'AMapLocation-NO-IDFA'
6  s.dependency 'AMapNavi-NO-IDFA'
7  s.dependency 'AMapSearch-NO-IDFA'
```

这样就指明了依赖的库里面包含有静态库, 否则就会报工程无法找到包含的静态库的错误, 如下图:

```
- ERROR | [iOS] unknown: Encountered an unknown error (The 'Pods-App' target
has transitive dependencies that include static binaries: (/private/var/folders/0
h/f80399_j7tv3gqvc38cgnhf40000gn/T/CocoaPods-Lint-20180614-95914-1mgiw7c-
/Pods/AMapLocation-NO-IDFA/AMapLocationKit.framework, /private/var/folders
/0h/f80399_j7tv3gqvc38cgnhf40000gn/T/CocoaPods-Lint-20180614-95914-1mgiw7c-
/Pods/AMapNavi-NO-IDFA/AMapNavikit.framework, and /private/var/folders/0
h/f80399_j7tv3gqvc38cgnhf40000gn/T/CocoaPods-Lint-20180614-95914-1mgiw7c-
/Pods/AMapSearch-NO-IDFA/AMapSearchKit.framework)) during validation.

[!] did not pass validation, due to 1 error.
You can use the '--no-clean' option to inspect any issue.

[!] 'AMap3DMap-NO-IDFA' uses the unencrypted http protocol to transfer the Pod. P
lease be sure you're in a safe network with only trusted hosts in there. Please r
each out to the library author to notify them of this security issue.

[!] 'AMapFoundation-NO-IDFA' uses the unencrypted http protocol to transfer the P
od. Please be sure you're in a safe network with only trusted hosts in there. Ple
ase reach out to the library author to notify them of this security issue.
```



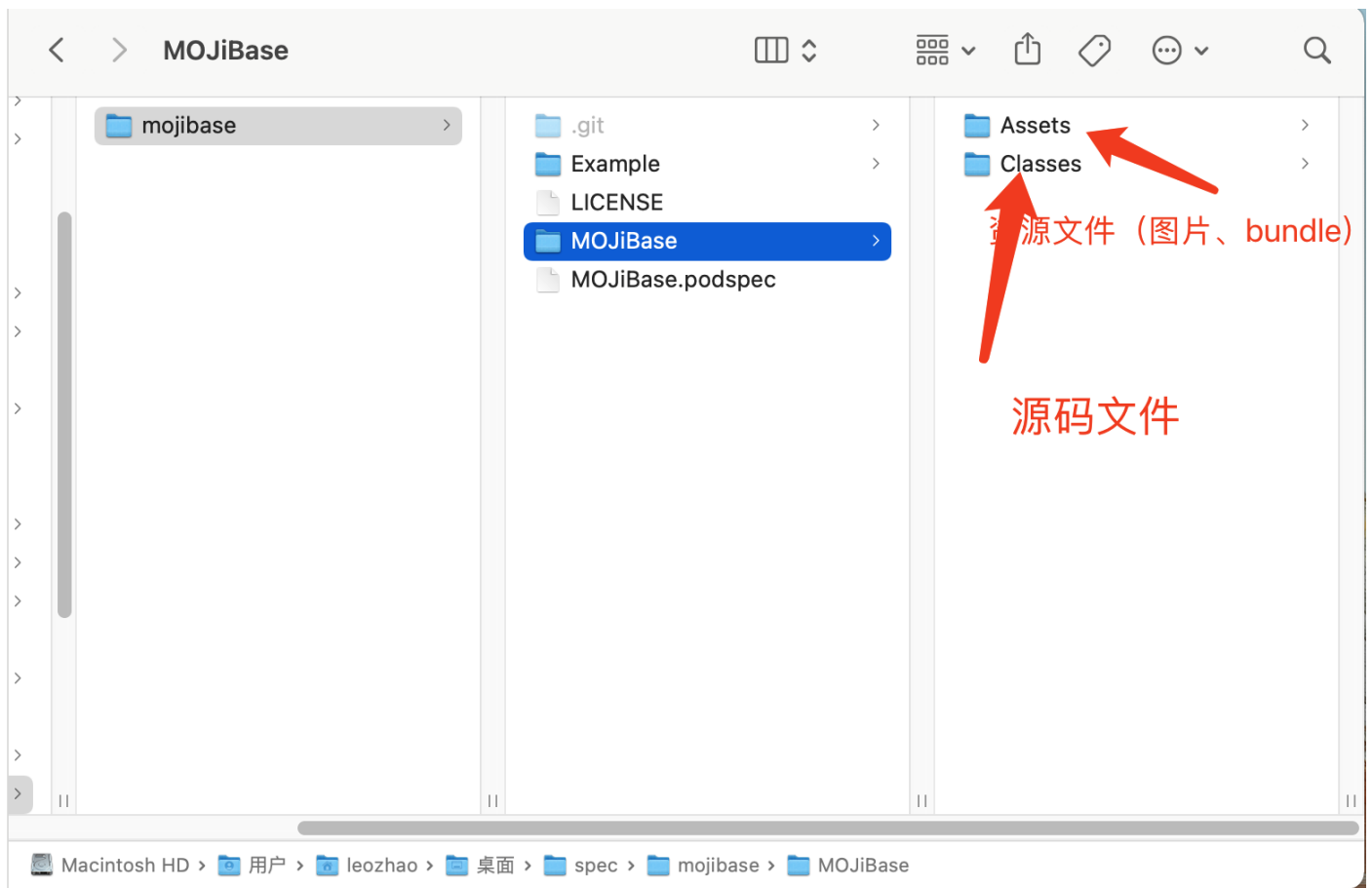
更多踩坑指南请访问：<https://xiaohuicoding.github.io/2018/03/01/CocoaPods%E5%88%9B%E5%BB%BA%E7%A7%81%E6%9C%89%E5%BA%93%E7%BC%88%E8%B8%A9%E5%9D%91%E7%AF%87%E7%BC%89/>

6、添加源码和资源文件：

如图所示：

Classes存放所有的代码文件

Assets存放所有的资源文件



1、访问图片资源、bundle、xib等文件

1、因为pod会将我们的bundle、xib、png等资源文件打包进pod项目的主Bundle中的Base.framework中,所以不能用以前的方式直接读取图片，而且实用NSBundle.mainBundle也容易和主项目产生冲突。

OC版本：新建NSBundle+Extension分类

Objective-C

```
1 + (NSBundle *)bundleWithBundleName:(NSString *)bundleName podName:(NSString *)
  podName {
2     if (bundleName == nil && podName == nil) {
3         @throw @"bundleName和podName不能同时为空";
4     }else if (bundleName == nil ) {
5         bundleName = podName;
6     }else if (podName == nil) {
7         podName = bundleName;
8     }
9
10
11     if ([bundleName containsString:@"bundle"]) {
12         bundleName = [bundleName componentsSeparatedByString:@"bundle"].firstObje
            ct;
13     }
14     //没使用framework的情况下
15     NSURL *associateBundleURL = [[NSBundle mainBundle] URLForResource:bundleName
            withExtension:@"bundle"];
16     //使用framework形式
17     if (!associateBundleURL) {
18         associateBundleURL = [[NSBundle mainBundle] URLForResource:@"Frameworks" w
            ithExtension:nil];
19         associateBundleURL = [associateBundleURL URLByAppendingPathComponent:podNa
            me];
20         associateBundleURL = [associateBundleURL URLByAppendingPathExtension:@"fra
            mework"];
21         NSBundle *associateBunle = [NSBundle bundleWithURL:associateBundleURL];
22         associateBundleURL = [associateBunle URLForResource:bundleName withExtensi
            on:@"bundle"];
23     }
24
25     NSAssert(associateBundleURL, @"取不到关联bundle");
26     //生产环境直接返回空
27     return associateBundleURL?[NSBundle bundleWithURL:associateBundleURL]:nil;
28 }
```

 Swift版本

```

1  extension Bundle {
2
3      class func fetchBundle(bundleName: String?, podName: String?) -> Bundle? {
4          var bundleName = bundleName
5          var podName = podName
6          //判断参数
7          if bundleName == nil && podName == nil {
8              assert((bundleName != nil), "bundleName 和 podName不能同时为空")
9          } else if (bundleName == nil) {
10              bundleName = podName
11          } else if (podName == nil) {
12              podName = bundleName
13          }
14          //适配名称
15          if ((bundleName?.contains(".bundle")) != nil) {
16              bundleName = bundleName?.components(separatedBy: ".bundle").first
17          }
18          var associateBundleURL = Bundle.main.url(forResource: bundleName,
withExtension: "bundle")
19          //如果associateBundleURL为空, 则走下面方法
20          if associateBundleURL == nil {
21              associateBundleURL = Bundle.main.url(forResource: "Frameworks",
withExtension: nil)
22              associateBundleURL =
associateBundleURL?.appendingPathComponent(podName!)
23              associateBundleURL =
associateBundleURL!.appendingPathExtension("framework")
24              let associateBunle = Bundle(url: associateBundleURL!)
25              associateBundleURL = associateBunle?.url(forResource: bundleName,
withExtension: "bundle")
26          }
27          assert((associateBundleURL != nil), "取不到关联bundle")
28
29          return Bundle(url: associateBundleURL!) ?? nil
30
31      }
32  }

```

7、验证podspec文件是否无误

1、主要使用命令

cd 到 MOJiBase项目文件夹下。其目录下有 MOJiBase.podspec 文件。

执行以下命令：

Apache

```
1 pod lib lint --allow-warnings
2 #严谨一点，可以不忽略警告⚠️，如下
3 pod lib lint
```

如果验证成功将会显示如下图：

```
er APIs such as CTFontCreateUIFontForLanguage() or +[UIFont systemFontOfSize:].
- NOTE | [iOS] xcodebuild: 2021-09-16 11:20:47.164 IBAgent-iOS[83591:28425615] CoreText note: Set a breakpoint on CTFo
ntLogSystemFontNameRequest to debug.
- NOTE | [iOS] xcodebuild: 2021-09-16 11:20:47.164 IBAgent-iOS[83591:28425615] CoreText note: Client requested name ".
SFUI-Semibold", it will get TimesNewRomanPSMT rather than the intended font. All system UI font access should be through pro
per APIs such as CTFontCreateUIFontForLanguage() or +[UIFont systemFontOfSize:].
- NOTE | [iOS] xcodebuild: 2021-09-16 11:20:47.164 IBAgent-iOS[83591:28425615] CoreText note: Client requested name ".
SFUI-RegularItalic", it will get TimesNewRomanPSMT rather than the intended font. All system UI font access should be throug
h proper APIs such as CTFontCreateUIFontForLanguage() or +[UIFont systemFontOfSize:].
- NOTE | [iOS] xcodebuild: 2021-09-16 11:20:47.231 IBAgent-iOS[83592:28425843] CoreText note: Client requested name ".
SFUI-Regular", it will get TimesNewRomanPSMT rather than the intended font. All system UI font access should be through prop
er APIs such as CTFontCreateUIFontForLanguage() or +[UIFont systemFontOfSize:].
- NOTE | [iOS] xcodebuild: 2021-09-16 11:20:47.231 IBAgent-iOS[83592:28425843] CoreText note: Set a breakpoint on CTFo
ntLogSystemFontNameRequest to debug.
- NOTE | [iOS] xcodebuild: 2021-09-16 11:20:47.231 IBAgent-iOS[83592:28425843] CoreText note: Client requested name ".
SFUI-Semibold", it will get TimesNewRomanPSMT rather than the intended font. All system UI font access should be through pro
per APIs such as CTFontCreateUIFontForLanguage() or +[UIFont systemFontOfSize:].
- NOTE | [iOS] xcodebuild: 2021-09-16 11:20:47.231 IBAgent-iOS[83592:28425843] CoreText note: Client requested name ".
SFUI-RegularItalic", it will get TimesNewRomanPSMT rather than the intended font. All system UI font access should be throug
h proper APIs such as CTFontCreateUIFontForLanguage() or +[UIFont systemFontOfSize:].
- NOTE | [iOS] xcodebuild: warning: Skipping code signing because the target does not have an Info.plist file and one
is not being generated automatically. (in target 'App' from project 'App')
- NOTE | [iOS] xcodebuild: warning: no rule to process file '/Users/leozhao/Desktop/spec/mojibase/M0JiBase/Classes/Lan
gTools/ZHTools/simp2trad.json' of type 'text.json' for architecture 'arm64' (in target 'M0JiBase' from project 'Pods')
- NOTE | [iOS] xcodebuild: warning: no rule to process file '/Users/leozhao/Desktop/spec/mojibase/M0JiBase/Classes/Lan
gTools/ZHTools/simp2trad.json' of type 'text.json' for architecture 'x86_64' (in target 'M0JiBase' from project 'Pods')

M0JiBase passed validation.
leozhao@leodeMac-mini mojibase %
```

如果报Error可以用如下命令查看详细错误信息：

Bash

```
1 pod lib lint --verbose
```

2、常见的几个警告⚠️信息

1、summary: The summary is not meaningful.

解释：这是因为你没有修改 `.podspec` 文件中的 `s.summary` 字段。只需要修改 `.podspec` 文件即可，如下：

Bash

```
1 s.summary = '这是一个测试组件库'
```

2、swift: The validator used Swift 3.2 by default because no Swift version was specified

解释：没有指定Swift版本，只需要在 `.podspec` 增加对swift版本指定即可，如下：

Plain Text

```
1 s.swift_version = '5.0'
```



如果两个不同的 target 包含相同的依赖关系，但需要不同的 Swift 版本，则 CocoaPods 现在将为同一 Pod 创建多个不同的目标，以适应所使用的不同 Swift 版本。如下：

Bash

```
1 Pod::Spec.new do |s|
2   s.name = 'MOJiBase'
3   s.version = '1.3.11'
4   # ...rest of attributes here
5   s.swift_versions = ['4.0', '5.0']
6 end
```

解决完上面两个WARN之后，再执行 `pod lib lint --allow-warnings` 命令就会发现没有警告了。



后续欢迎补充更多“常见错误”以帮助开发小伙伴避坑！

8、提交MOJiBase到gitlab仓库

cd到mojibase的本地文件目录下，按照如下步骤进行git操作：

Apache

```
1 git add --all
2 git commit -a -m "第一次提交"
3 git push origin master
4 git tag 1.3.11
5 git push --tags
```

9、将podspec提交到gitlab

cd 到 MOJiBase项目文件夹下。其目录下有MOJiBase.podspec 文件。运行如下命令：

Apache

```
1 pod repo push MOJiBase MOJiBase.podspec --allow-warnings
2 #命令解释
3 #pod repo push 【私有库名称】 【podspec文件名】 --allow-warnings
```

上传成功后将会显示如下信息：

```
Validating spec
-> B0TestTools (0.1.0)
  - NOTE | xcodebuild: note: Using new build system
  - NOTE | [iOS] xcodebuild: note: Planning build
  - NOTE | [iOS] xcodebuild: note: Constructing build description
  - NOTE | [iOS] xcodebuild: warning: Skipping code signing because the target does not have an Info.plist file. (in target 'App')

Updating the `B0TestSpec' repo

Already up to date.

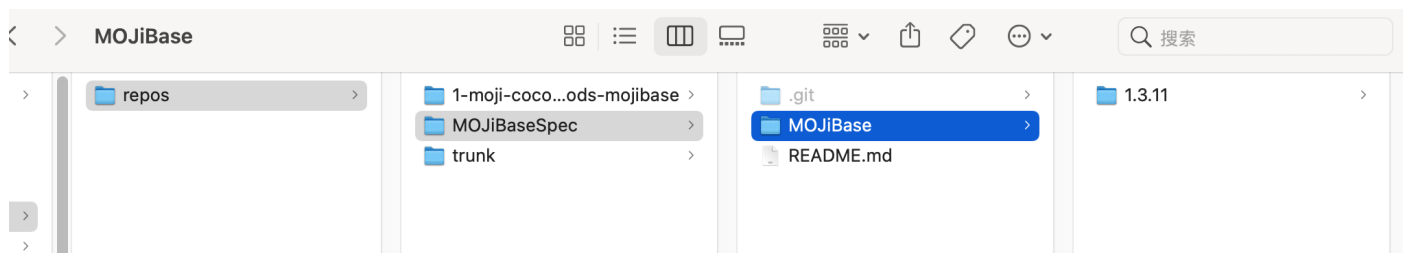
Adding the spec to the `B0TestSpec' repo

- [Add] B0TestTools (0.1.0)

Pushing the `B0TestSpec' repo
```

@掘金技术社区

显示如上信息后，你还可以查看 ~/.cocoapods/repos。在 MOJiBaseSpec 仓库下会多新增 MOJiBase 文件夹。如下所示：



🎉 至此，你的私有库已制作完成。你和你的小伙伴们可以在项目中使用它了。

10、在你的项目中使用你的私有库

1、创建MOJiBaseTest测试项目。

2、在项目目录下，执行命令：

Plain Text


```
1 pod init
```

3、编辑 podfile 文件：

不仅需要pod我们需要的私有库，还需要pod私有库依赖的公共库
依赖的公共库如图所示：

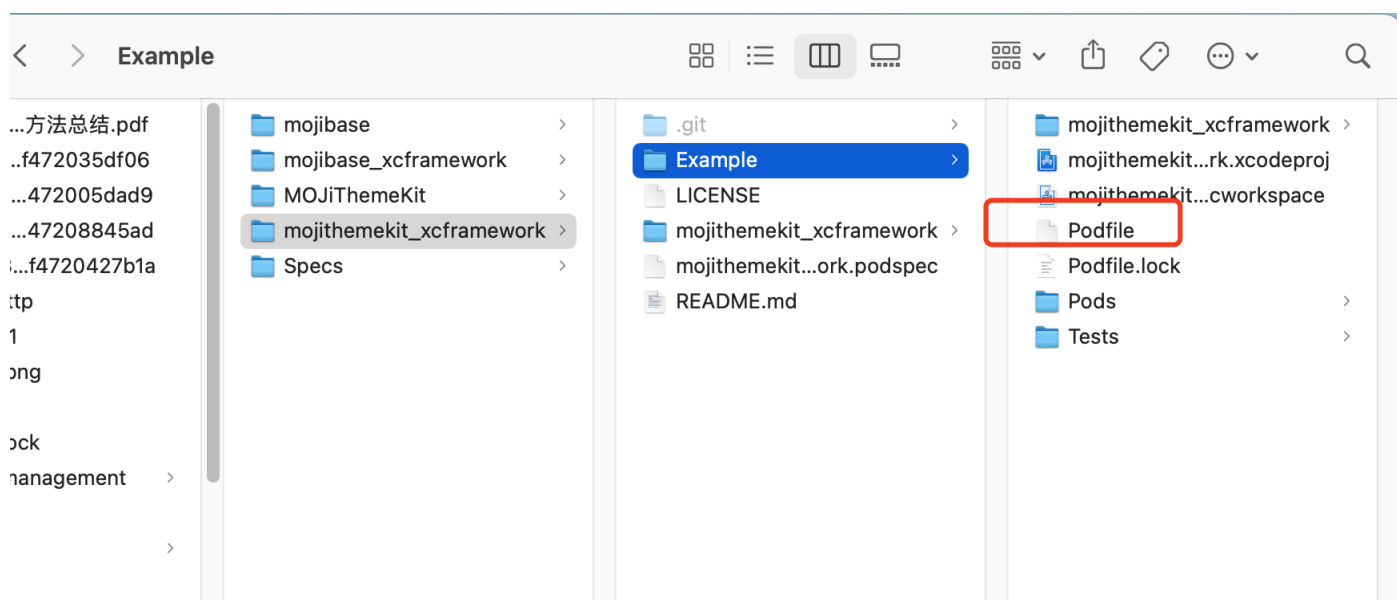
Bash

```
1 s.dependency 'SVProgressHUD'  
2 s.dependency 'SwiftyJSON'  
3 s.dependency 'SnapKit'
```

 注意，如果依赖的第三方框架里有我们自己的私有库，那么需要在Example文件下的podfile中增加我们的source源，并在podspec文件中写明依赖关系，如图：
(source源将在下面👉的内容讲到。)

Bash

```
1 s.dependency 'MOJiBase'
```



Podfile

```
use_frameworks!

platform :ios, '11.0'
|
source 'https://cdn.cocoapods.org/'
source 'http://192.168.1.3/moji-cocoapods/Specs.git'

target 'mojithemekit_xcframework_Example' do
  pod 'mojithemekit_xcframework', :path => '../'

  target 'mojithemekit_xcframework_Tests' do
    inherit! :search_paths
  end
end
```

Podfile的几种写法

👉 官方podfile写法指导: <https://guides.cocoapods.org/using/the-podfile.html>

1、写source

❌ source '<https://github.com/CocoaPods/Specs.git>'国内基本不可用，开网络代理也没用，所以需要使用国内映射，如下：

✅ cocoapods国内映射: source '<https://cdn.cocoapods.org/>'
(国内cdn映射，具体信息详见cocoapods官网: <https://blog.cocoapods.org/CocoaPods-1.7.2/>)

Bash

```
1 platform :ios, '11.0'
2 #国内基本不可用，开网络代理也没用，所以需要使用国内映射
3 #source 'https://github.com/CocoaPods/Specs.git' # cocoapods官方库
4 #国内cdn映射，cocoapods官网认证: https://blog.cocoapods.org/CocoaPods-1.7.2/
5 source 'https://cdn.cocoapods.org/'
6 source 'http://192.168.1.3/moji-cocoapods/Specs.git' #MOJi私有库Repo地址
7 target 'MOJiBaseTest' do
8   # Comment the next line if you're not using Swift and don't want to use
   dynamic frameworks
9   use_frameworks!
10  # Pods for MOJiBaseTest
11  pod 'MOJiBase'
12  pod 'SVProgressHUD'
13  pod 'SwiftyJSON'
14  pod 'SnapKit'
15 end
16
```



注意：私有仓库的地址一定是Spec Repo 的地址，不要错误的使用MOJiBase组件的git仓库。

2、不写source，需明确pod地址和tag

Ruby

```
1 platform :ios, '11.0'
2
3 target 'MOJiBaseTest' do # Comment the next line if you're not using Swift and
   don't want to use dynamic frameworks
4   use_frameworks!
5   # Pods for MOJiBaseTest
6   pod 'MOJiBase', :git => 'http://192.168.1.3/moji-cocoapods/mojibase.git', :tag
   => '1.3.11'
7   pod 'SVProgressHUD'
8   pod 'SwiftyJSON'
9   pod 'SnapKit'
10  end
```

然后执行 `pod install`，添加私有库到工程中。显示如下图则为成功。

```
test — -zsh — 80x24

Pod installation complete! There are 3 dependencies from the Podfile and 3 total
pods installed.

[!] Smart quotes were detected and ignored in your Podfile. To avoid issues in t
he future, you should not use TextEdit for editing it. If you are not using Text
Edit, you should turn off smart quotes in your editor of choice.
leozhao@leodeMac-mini test % pod install
/System/Library/Frameworks/Ruby.framework/Versions/2.6/usr/lib/ruby/2.6.0/univer
sal-darwin20/rbconfig.rb:229: warning: Insecure world writable dir /opt/homebrew
/sbin in PATH, mode 040777
Analyzing dependencies
Pre-downloading: `MOJIBase` from `http://192.168.1.3/moji-cocoapods/mojibase.git
` tag `1.3.11`
Downloading dependencies
Installing MOJIBase (1.3.11)
Generating Pods project
Integrating client project
Pod installation complete! There are 4 dependencies from the Podfile and 4 total
pods installed.

[!] Smart quotes were detected and ignored in your Podfile. To avoid issues in t
he future, you should not use TextEdit for editing it. If you are not using Text
Edit, you should turn off smart quotes in your editor of choice.
leozhao@leodeMac-mini test %
```



至此，cocoapods私有库创建完成，以下为cocoapods总结的其他使用方法总结，有兴趣的可以继续往下看。

11、其他扩展使用

spec文件中支持 `Debug` 和 `Release` 的区分设置

Bash

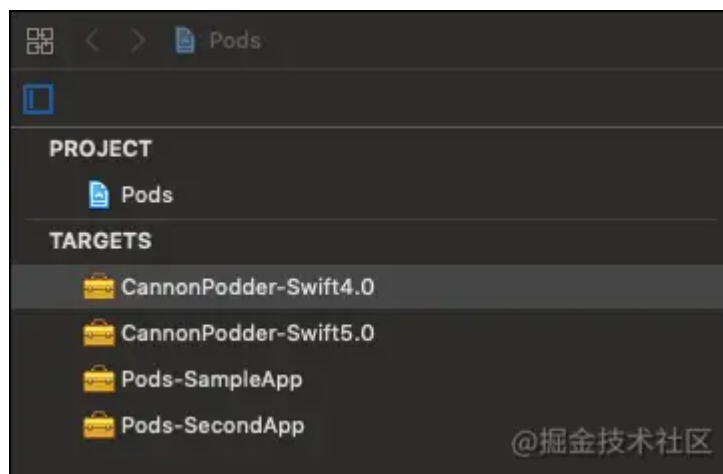
```
1 s.dependency 'ButterDebugging', :configurations => ['Debug']
2 s.dependency 'ErrorReportingTool', :configurations => ['Release']
```

以及以下的Podfile

Delphi

```
1 target 'SampleApp'
2     do supports_swift_version '< 5.0'
3     pod 'CannonPodder'
4 end
5
6 target 'SecondApp'
7     do supports_swift_version '>= 5.0'
8     pod 'CannonPodder'
9 end
10
```

将创建 CannonPodder 目标的两个不同版本。



这样可以确保为依赖于 Pod 的每个目标使用正确的 Swift 版本。

XCFramework打包步骤

podspec文件写法：

Ruby

```
1 #
2 # Be sure to run `pod lib lint MOJiBase_XCFramework.podspec' to ensure this is
  a
3 # valid spec before submitting.
4 #
5 # Any lines starting with a # are optional, but their use is encouraged
6 # To learn more about a Podspec see
  https://guides.cocoapods.org/syntax/podspec.html
7 #
8
9 Pod::Spec.new do |s|
10   s.name          = 'MOJiBase_XCFramework'
11   s.version       = '1.3.11'
12   s.summary       = 'MOJiBase_XCFramework'
13   #s.vendored_frameworks使用任何一句都可以
14   #s.vendored_frameworks = 'MOJiBase_XCFramework/Classes/MOJiBase.xcframework'
15   s.vendored_frameworks = 'MOJiBase_XCFramework/**/*.{xcframework}'
16   s.swift_version = '5.0'
17   # This description is used to generate tags and improve search results.
18   # * Think: What does it do? Why did you write it? What is the focus?
19   # * Try to keep it short, snappy and to the point.
20   # * Write the description between the DESC delimiters below.
21   # * Finally, don't worry about the indent, CocoaPods strips it!
22
23   s.description    = <<-DESC
24   TODO: Add long description of the pod here.
25       DESC
26
27   s.homepage      = 'http://192.168.1.3/moji-cocoapods/mojibase\_xcframework.git'
28   # s.screenshots  = 'www.example.com/screenshots\_1',
  www.example.com/screenshots\_2'
29   s.license       = { :type => 'MIT', :file => 'LICENSE' }
30   s.author        = { 'zhaohanjun' => 'zhaohanjun@mail.shareintelli.com' }
31   s.source        = { :git =>
  'http://192.168.1.3/moji-cocoapods/mojibase\_xcframework.git', :tag =>
  s.version.to_s }
32   # s.social_media_url = 'https://twitter.com/<TWITTER\_USERNAME>'
33   #s.source_files = 'MOJiBase_XCFramework/Classes/**/*.h'
34   s.ios.deployment_target = '11.0'
35   s.dependency    'SVProgressHUD'
36   s.dependency    'SwiftyJSON'
37   s.dependency    'SnapKit'
38 end
```



⚠️注意项，主要差别体现在下面这句配置

Bash

```
1 #源码和xcframework的配置信息有所不同，主要体现在下面这句配置
2 #声明xcframework的依赖地址
3 s.vendored_frameworks = 'MOJiBase_XCFramework/Classes/MOJiBase.xcframework'
```

如果出现以下错误可以在pod命令后加上 `--skip-import-validation`，如图所示：

```
(2 failures)
Testing with `xcodebuild`.
-> MOJiBase_XCFramework (1.3.11)
- ERROR | [iOS] xcodebuild: Returned an unsuccessful exit code.
- NOTE | xcodebuild: note: Using new build system
- NOTE | xcodebuild: note: Building targets in parallel
- NOTE | xcodebuild: note: Using codesigning identity override: -
- NOTE | xcodebuild: note: Build preparation complete
- NOTE | [iOS] xcodebuild: note: Planning build
- NOTE | [iOS] xcodebuild: note: Analyzing workspace
- NOTE | [iOS] xcodebuild: note: Constructing build description
- NOTE | [iOS] xcodebuild: Pods.xcodeproj: warning: The iOS Simulator deployment target 'IPHONEOS_DEPLOYMENT_TARGET' is set to 8.0, but the range of supported deployment target versions is 9.0 to 14.5.99. (in target 'SVProgressHUD' from project 'Pods')
- NOTE | [iOS] xcodebuild: warning: [CP] Unable to find matching .xcframework slice in '...../Users/leozhao/Desktop/spec/mojibase_xcframework/MOJiBase_XCFramework/Classes/MOJiBase.xcframework MOJiBase framework ios-arm64 ios-x86_64-simulator' for the current build architectures (arm64 x86_64).
- NOTE | xcodebuild: ld: warning: directory not found for option '-F/Users/leozhao/Library/Developer/Xcode/DerivedData/App-gfxaeddvdldrvodettodyursajent/Build/Products/Release-iponesimulator/XCFrameworkIntermediates/MOJiBase'
- NOTE | xcodebuild: clang: error: linker command failed with exit code 1 (use -v to see invocation)
- NOTE | [iOS] xcodebuild: warning: Skipping code signing because the target does not have an Info.plist file and one is not being generated automatically. (in target 'App' from project 'App')

[!] MOJiBase_XCFramework did not pass validation, due to 1 error.
```

则执行下面的方法

Visual Basic

```
1 pod lib lint --verbose --skip-import-validation
```

```
Support Files/Pods-App/Pods-App-dummy.m
- Integrating targets

Integrating target `MOJiBase_XCFramework`
  Adding Build Phase '[CP] Copy XCFrameworks' to project.
  - Stabilizing target UUIDs
  - Running post install hooks
  - Writing Xcode project file to
    `../../../../../../../../private/var/folders/7q/tv965qr14m329909pf77dlth0000gn/T/CocoaPods-Lint-20
210918-46898-1etyqzh-MOJiBase_XCFramework/Pods/Pods.xcodeproj`
  Cleaning up sandbox directory

Integrating client project

[!] Please close any current Xcode sessions and use `App.xcworkspace` for this project from
now on.

Integrating target `Pods-App` (`../../../../../../../../var/folders/7q/tv965qr14m329909pf77dlth0000g
n/T/CocoaPods-Lint-20210918-46898-1etyqzh-MOJiBase_XCFramework/App.xcodeproj` project)
  Adding Build Phase '[CP] Embed Pods Frameworks' to project.
  Adding Build Phase '[CP] Check Pods Manifest.lock' to project.
  - Running post integrate hooks

-> Pod installation complete! There is 1 dependency from the Podfile and 4 total pods insta
lled.
  Building with `xcodebuild`.
  Testing with `xcodebuild`.
-> MOJiBase_XCFramework (1.3.11)

MOJiBase_XCFramework passed validation.
[!] Skipping compilation with `xcodebuild` because target contains no sources.

leozhao@leodeMac-mini mojibase_xcframework %
```

✓ 完成验证

第一步

✓ MOJiBase_XCFramework 必须同项目名相同

Bash

```
1 pod repo add MOJiBase_XCFramework http://192.168.1.3/moji-cocoapods/Specs.git
```

第二步

Assembly language

```
1 pod repo push MOJiBase_XCFramework MOJiBase_xcframework.podspec --allow-warnin
gs --skip-import-validation
```

第三步

因为我们用上面两部会产生重复的spec文件，所以我们需要到“~/cocoapods/repos”文件下，删除多余的spec目录，保留一个即可。然后在最后的spec目录“git pull origin master” / “pod repo update” 一下。

目前还没有找到其他的方法解决这个问题，等我弄清楚，将会更新这个文件。

或者

不用上面那么麻烦，将所有的spec文件移动到“Spec”文件下，然后执行下面的方法：

Bash

```
1 pod repo push Spec MOJiBase_xcframework.podspec --allow-warnings --skip-import-validation
```



更多cocoapods私有库配置避坑指南参考：http://www.mengyueping.com/2018/08/16/iOS_CocoaPods_03/

终