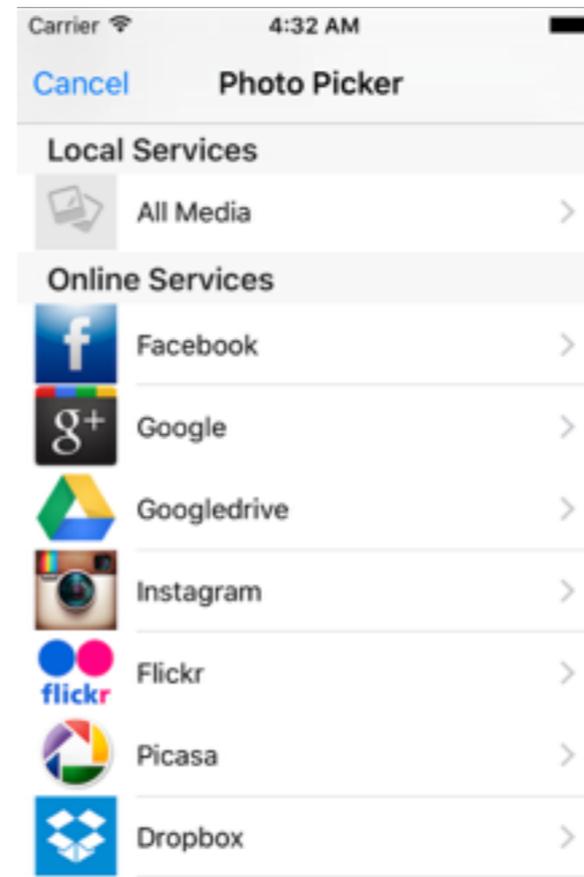
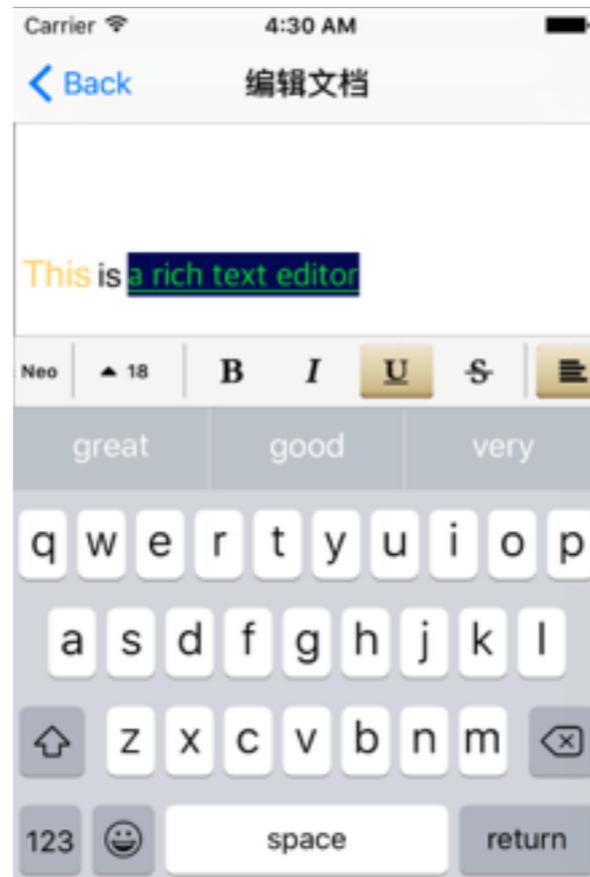
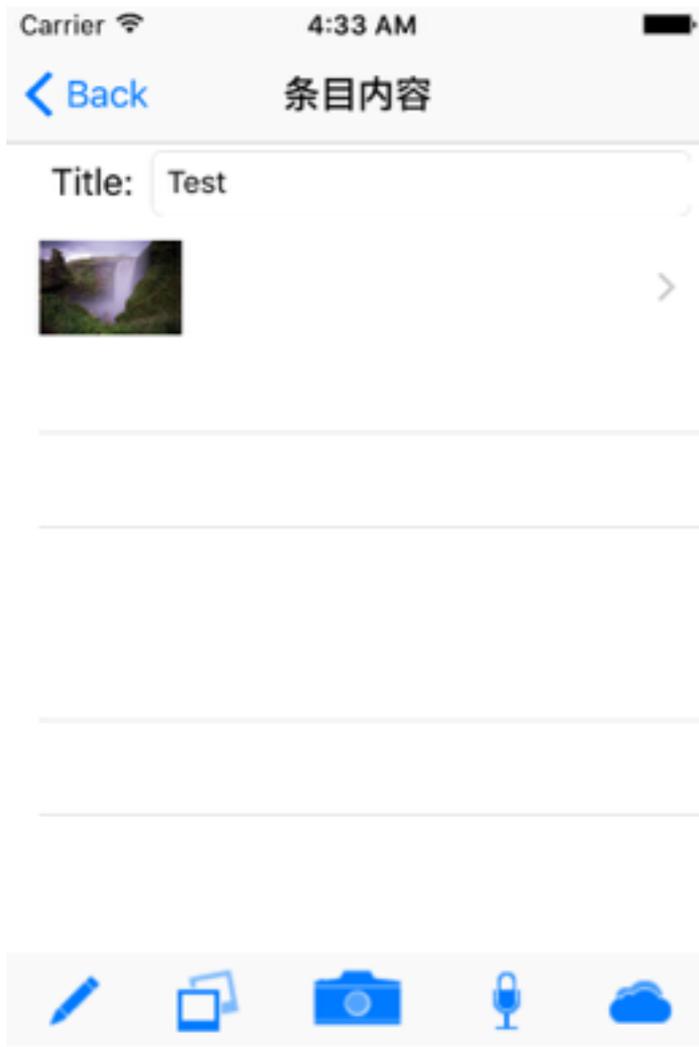


CocoaPods 模块管理

Wu Yang
@pinxue
2015

重用的威力



开源库的使用

JSONModel

PhotoPickerPlus

CLImageEditor

SDWebImage

GPUImage

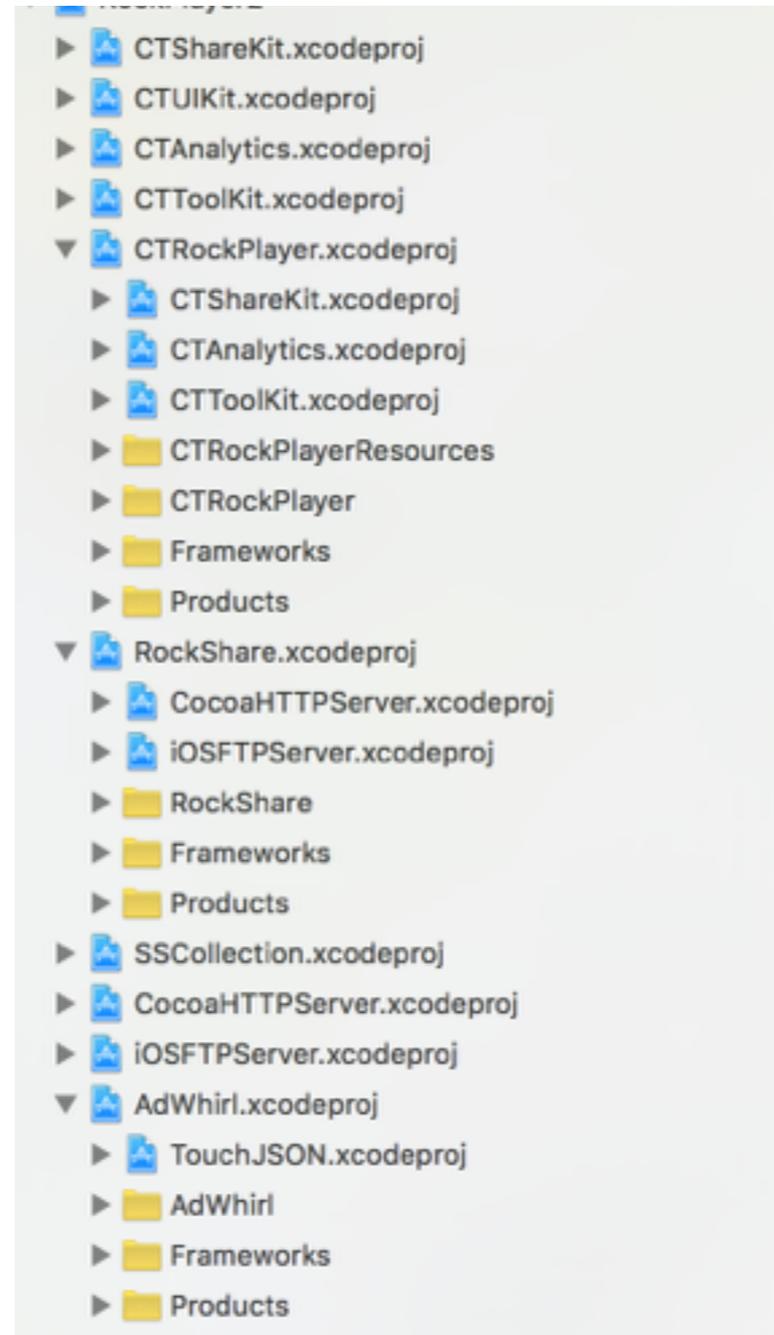
iOS-Rich-Text-Editor

Novocaine

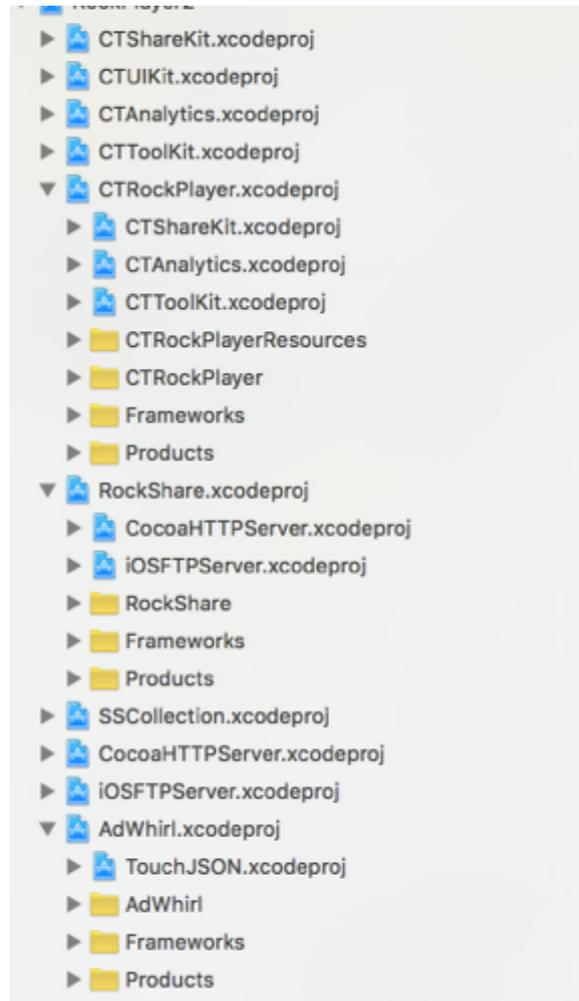
EZAudio

挑选和解决异常花的时间远比使用要多

内部模块



工程管理的麻烦



Xcode workspace
Static Library
iOS Framework
git/hg subrepo

模块化的挑战

- 1 个工程会使用很多模块
- 模块最好像积木一样，拿来就用
- 最好不用折腾文件搜索路径
- 最好不要手工复制文件
- 找开源库要容易
- 内部模块跟外来模块能完美共存
- 模块版本要能随时更新
- 模块不仅提供代码，还能提供资源

Xcode workspace 提供的:

- with source code
 - group + external file ref
 - project dep
- with binary and header files
 - static lib
 - static framework (unofficial)
 - embed framework (iOS8+)

CocoaPods History

<https://github.com/CocoaPods/CocoaPods/blob/master/CHANGELOG.md>

0.39.0 (2015-10-09)

<http://blog.cocoapods.org/CocoaPods-0.38/> watchos, targets dedup

<http://blog.cocoapods.org/CocoaPods-0.37/> pod cache, module mapping
<http://clang.llvm.org/docs/Modules.html#module-maps>

<http://blog.cocoapods.org/CocoaPods-0.36/> swift, framework

<http://blog.cocoapods.org/CocoaPods-0.35/> new molinillo dep resolver

<http://blog.cocoapods.org/CocoaPods-0.34/> configuration dep, private spec repo

<http://blog.cocoapods.org/CocoaPods-0.33/> trunk, plugins command

<http://blog.cocoapods.org/CocoaPods-0.32/> update single pod, spec.deprecated

<http://blog.cocoapods.org/CocoaPods-0.31/>

<http://blog.cocoapods.org/CocoaPods-0.30/>

<http://blog.cocoapods.org/CocoaPods-0.29/>

<http://blog.cocoapods.org/CocoaPods-0.28/>

<http://blog.cocoapods.org/CocoaPods-0.27-and-improved-installation-UX/>

<http://blog.cocoapods.org/CocoaPods-0.26/>

有 Carthage 等竞争.....

CocoaPods Basic

CocoaPods Is

- a dependency management tool
 - for Xcode
 - written in Ruby
 - distributed via RubyGems
 - invoked from command line
- a package distributing system
 - using a Spec.git as category
 - with official repo hosted on github.com
 - and support private repo

Github, PR, BDD

Running the specs

1019 specifications (1645 requirements)

Running Integration tests

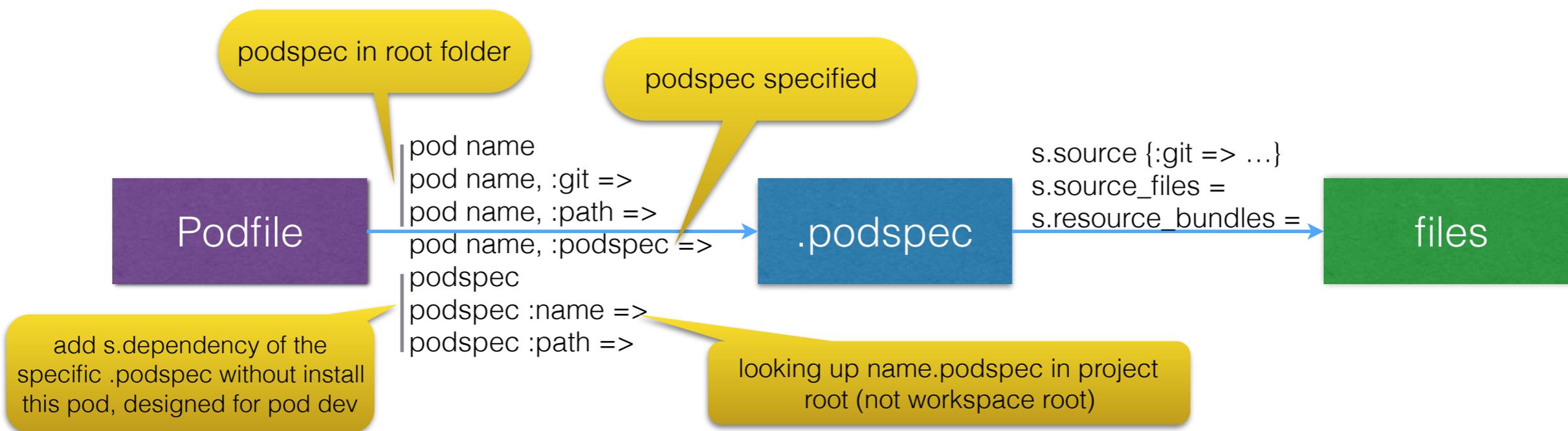
850 specifications (1650 requirements)

Running examples

Running RuboCop

Running Inch

Pod: any fileset with .podspec



- A pod is defined by its podspec
 - User add a Podfile in project to declare the library needed.
 - Podfile refers the pods by their .podspec
 - .podspec can be anywhere
 - in the root folder of a git repo (master or private server or local)
 - explicit provided by :podspec parameter
 - even from repo of dependency of a podspec (podspec command in Podfile)
- The .podspec can reference files anywhere
 - a public github project
 - a internal project over the private git server, maybe powered by gitlab
 - a git repo in local file system
 - some files available over the network, regardless in a git repo or not
 - some files in local file system

"podspec" directive in Podfile

0.36+ requires source

Finding Podfile changes
A SimplePanel
- MyPod3

Fetching external sources
-> Fetching podspec for `MyPod3` from `../`

Resolving dependencies of `Podfile`

Comparing resolved specification to the sandbox manifest
A SimplePanel
- MyPod3

Downloading dependencies

-> Using MyPod3 (0.1.0)

-> Installing SimplePanel (0.1)

> Git download
\$ /usr/bin/git clone https://github.com/azu/SimplePanel.git /var/folders/vp/... --single-branch --depth 1 --branch 0.1
> Copying SimplePanel from `~/Library/Caches/CocoaPods/Pods/Release/SimplePanel/0.1-4afe1` to `Pods/SimplePanel`
- Running pre install hooks

Generating Pods project

- Creating Pods project
- Adding source files to Pods project
- Adding frameworks to Pods project
- Adding libraries to Pods project
- Adding resources to Pods project
- Linking headers
- Installing targets
- Installing target `MyPod3` iOS 7.1
 - Generating Info.plist file, module map file, umbrella header ...

- Installing target `SimplePanel` iOS 7.1
 - Generating Info.plist file at `Pods/Target Support Files/SimplePanel/Info.plist`
 - Generating module map file at `Pods/Target Support Files/SimplePanel/SimplePanel.modulemap`
 - Generating umbrella header at `Pods/Target Support Files/SimplePanel/SimplePanel-umbrella.h`

- Installing target `Pods-MyPod3_Example` iOS 7.1
 - Generating Info.plist file, module map file, umbrella header ...
- Installing target `Pods-MyPod3_Tests` iOS 7.1
 - Generating Info.plist file, module map file, umbrella header ...
- Running post install hooks
- Writing Xcode project file to `Pods/Pods.xcodeproj`
- Writing Lockfile in `Podfile.lock`
- Writing Manifest in `Pods/Manifest.lock`

Integrating client project

Integrating target `Pods-MyPod3_Example` (`MyPod3.xcodeproj` project)

Integrating target `Pods-MyPod3_Tests` (`MyPod3.xcodeproj` project)

- Running post install hooks
- cocoapods-stats from `~/Library/Ruby/Gems/2.0.0/gems/cocoapods-stats-0.5.3/lib/cocoapods_plugin.rb`

Sending stats

- SimplePanel, 0.1

```
Podfile in MyPod3 Example
source 'https://github.com/CocoaPods/Specs.git'
use_frameworks!
```

```
target 'MyPod3_Example', :exclusive => true do
  pod "MyPod3", :path => "../"
end
```

```
target 'MyPod3_Tests', :exclusive => true do
  pod "MyPod3", :path => "../"
  podspec :path => "/MyPrjs/FileLibrary/FileLibrary.podspec"
end
```

```
FileLibrary.podspec
Pod::Spec.new do |s|
  s.name           = "FileLibrary"
  ...
  s.dependency 'SimplePanel'
end
```

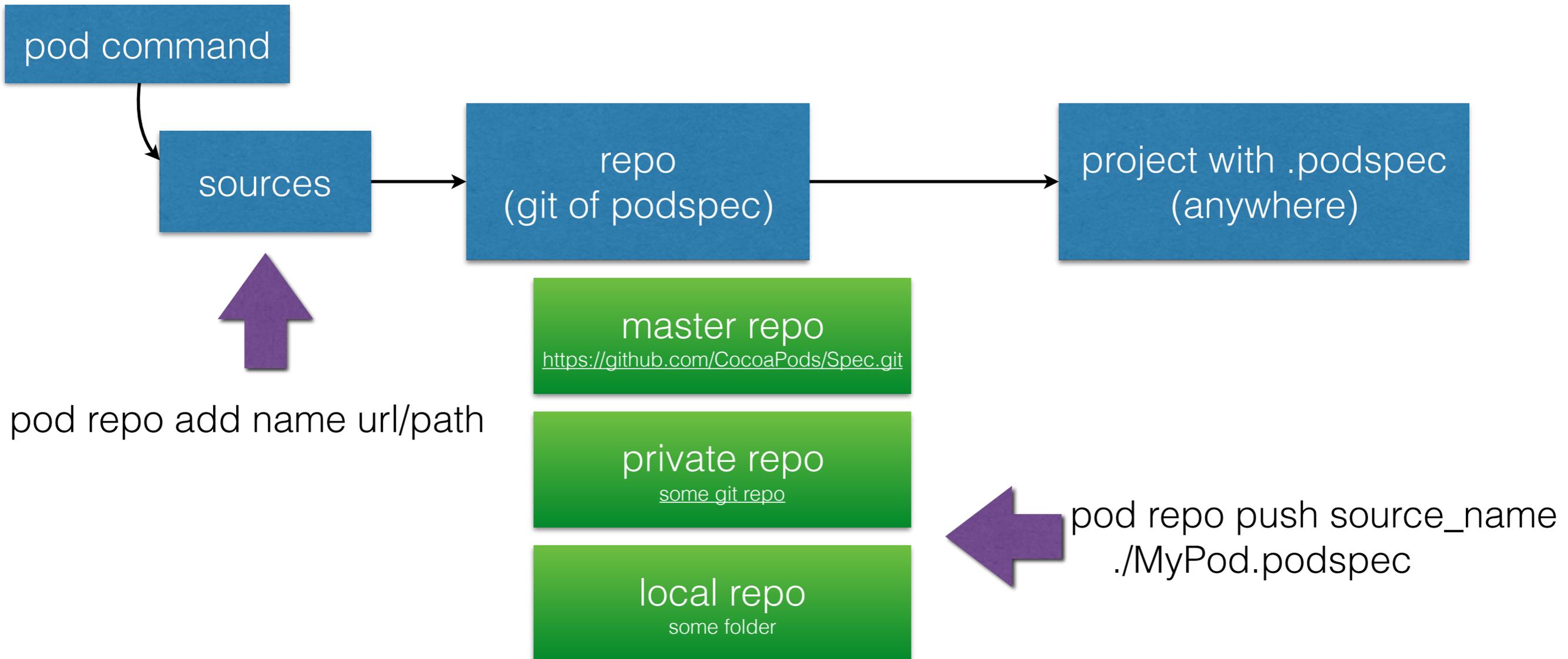
The screenshot shows the Xcode interface. On the left, the 'PROJECT' pane lists the 'Pods' folder and several targets: 'MyPod3', 'MyPod3-MyPod3', 'Pods-MyPod3_Example', 'Pods-MyPod3_Tests', and 'SimplePanel'. On the right, the 'Target Dependencies' pane for the selected target shows two dependencies: 'MyPod3 (Pods)' and 'SimplePanel (Pods)'. Below this, there are sections for 'Compile Sources (1 item)', 'Link Binary With Libraries (1 item)', and 'Headers (1 item)'.

Install

- Install CocoaPods in System
 - gem install cocoapods
 - pod command is available
 - pod setup
 - local repo is created ~/.cocoapods, with default source (master) <http://github.com/CocoaPods/Specs.git>
 - spec is cached, thus --no-update-repo works offline
 - podspec is shifting from ruby dsl to json, you will see .podspec.json instead of .podspec in cache
 - Files of pods ever used is cached in ~/Library/Caches/CocoaPods/Pods/
- Install CocoaPods in Project
 - create a Podfile along with .xcproject (pod init can help)
 - pod install --verbose

建议用 rvm

Repository



```
$ pod repo push --verbose phpods MyPod3.podspec
```

```
Validating spec
  MyPod3 (0.1.0) - Analyzing on iOS 7.0 platform.
  Preparing
```

```
Analyzing dependencies
```

```
Fetching external sources
-> Fetching podspec for `MyPod3` from `../MyPod3/MyPod3.podspec`
```

```
Resolving dependencies of
```

```
Comparing resolved specification to the sandbox manifest
  A MyPod3
```

```
Downloading dependencies
```

```
-> Installing MyPod3 (0.1.0)
> Git download
...
- Running pre install hooks
```

```
Generating Pods project
- Creating Pods project
- Adding source files to Pods project
- Adding frameworks to Pods project
- Adding libraries to Pods project
- Adding resources to Pods project
- Linking headers
- Installing targets
  - Installing target `MyPod3` iOS 7.0
    - Generating ... at ...
- Running post install hooks
- Writing Xcode project file to ...
- Writing Lockfile in ...
- Writing Manifest in ...
- Running post install hooks
```

```
Building with xcodebuild.
xcodebuild clean build -target Pods CODE_SIGN_IDENTITY=- -sdk iphonesimulator 2>&1
Build settings from command line:
  CODE_SIGN_IDENTITY = -
  SDKROOT = iphonesimulator9.0
```

```
=== CLEAN TARGET MyPod3-MyPod3 OF PROJECT Pods WITH THE DEFAULT CONFIGURATION (Release) ===
```

```
Check dependencies
```

```
Create product structure
/bin/mkdir -p /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.bundle
```

```
Clean.Remove clean /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.bundle
  builtin-rm -rf /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.bundle
```

```
Clean.Remove clean /var/folders/vp/.../CocoaPods/Lint/build/Pods.build/Release-iphonesimulator/MyPod3-MyPod3.build
  builtin-rm -rf /var/folders/vp/.../CocoaPods/Lint/build/Pods.build/Release-iphonesimulator/MyPod3-MyPod3.build
```

```
=== CLEAN TARGET MyPod3 OF PROJECT Pods WITH THE DEFAULT CONFIGURATION (Release) ===
```

```
Check dependencies
```

```
Create product structure
/bin/mkdir -p /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.framework
/bin/mkdir -p /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.framework/Headers
```

```
Clean.Remove clean /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.framework
  builtin-rm -rf /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.framework
```

```
Clean.Remove clean /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.framework.dSYM
  builtin-rm -rf /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.framework.dSYM
```

```
Clean.Remove clean /var/folders/vp/.../CocoaPods/Lint/build/Pods.build/Release-iphonesimulator/MyPod3.build
  builtin-rm -rf /var/folders/vp/.../CocoaPods/Lint/build/Pods.build/Release-iphonesimulator/MyPod3.build
```

pod repo push source_name MyPod3.podspec

```
=== CLEAN TARGET Pods OF PROJECT Pods WITH THE DEFAULT CONFIGURATION (Release) ===
```

```
Check dependencies
```

```
Create product structure
/bin/mkdir -p /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/Pods.framework
/bin/mkdir -p /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/Pods.framework/Headers
```

```
Clean.Remove clean /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/Pods.framework
  builtin-rm -rf /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/Pods.framework
```

```
Clean.Remove clean /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/Pods.framework.dSYM
  builtin-rm -rf /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/Pods.framework.dSYM
```

```
Clean.Remove clean /var/folders/vp/.../CocoaPods/Lint/build/Pods.build/Release-iphonesimulator/Pods.build
  builtin-rm -rf /var/folders/vp/.../CocoaPods/Lint/build/Pods.build/Release-iphonesimulator/Pods.build
```

```
** CLEAN SUCCEEDED **
```

```
=== BUILD TARGET MyPod3-MyPod3 OF PROJECT Pods WITH THE DEFAULT CONFIGURATION (Release) ===
```

```
Check dependencies
```

```
Create product structure
/bin/mkdir -p /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.bundle
```

```
Touch /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.bundle
  cd /var/folders/vp/.../CocoaPods/Lint/Pods
  export PATH="/Applications/Xcode-beta.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/usr/bin:/Applications/Xcode-beta.app/Contents/Developer/usr/bin:/Applications/Xcode-beta.app/Contents/Developer/usr/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin"
  /usr/bin/touch -c /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.bundle
```

```
=== BUILD TARGET MyPod3 OF PROJECT Pods WITH THE DEFAULT CONFIGURATION (Release) ===
```

```
Check dependencies
```

```
Write auxiliary files
/bin/mkdir -p /var/folders/vp/.../CocoaPods/Lint/build/Pods.build/Release-iphonesimulator/MyPod3.build/Objects-normal/x86_64
write-file /var/folders/vp/.../CocoaPods/Lint/build/Pods.build/Release-iphonesimulator/MyPod3.build/Objects-normal/x86_64/MyPod3.LinkFileList
...
```

```
Create product structure
/bin/mkdir -p /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.framework
/bin/mkdir -p /var/folders/vp/.../CocoaPods/Lint/build/Release-iphonesimulator/MyPod3.framework/Headers
...
** BUILD SUCCEEDED **
```

```
-> MyPod3 (0.1.0)
```

```
Updating the `phpods` repo
```

```
Already up-to-date.
```

```
Adding the spec to the `phpods` repo
```

```
$ /Applications/Xcode-beta.app/Contents/Developer/usr/bin/git status --porcelain
?? MyPod3/
- [Add] MyPod3 (0.1.0)
$ /Applications/Xcode-beta.app/Contents/Developer/usr/bin/git add MyPod3
$ /Applications/Xcode-beta.app/Contents/Developer/usr/bin/git commit --no-verify -m [Add] MyPod3 (0.1.0)
[master 5953624] [Add] MyPod3 (0.1.0)
 1 file changed, 33 insertions(+)
 create mode 100644 MyPod3/0.1.0/MyPod3.podspec
```

```
Pushing the `phpods` repo
```

```
To ssh://git@my.com/phpods.git
 9715474..5953624 master -> master
```

Create a pod

- pod lib
 - create
 - lint
- pod try
- pod repo push <repo> <.podspec>
- *Gitlab hint: .git name letter capital matters!
 - Use lowercase name in url
 - Even though camel case name can pass lint and you can use git with it
 - But pod push and pod install will fail!

CocoaPods mechanism

- static lib
 - since beginning
- framework (dynamic lib in bundle)
 - !use_frameworks
 - since 0.3.6 / iOS 8.0
 - 0.38 changed targets organising

Impl in 0.3.8

Step 0 - a simple project

single view project, with Unit Test and UI Test, no local git

PROJECT

 HelloCocoaPods

TARGETS

 HelloCocoaPods

 HelloCocoaPodsTests

 HelloCocoaPodsUITests

▶ Target Dependencies (0 items)

▶ Compile Sources (3 items)

▶ Link Binary With Libraries (0 items)

▶ Copy Bundle Resources (3 items)

```
Podfile
use_frameworks!
pod 'AFNetworking', '~> 2.0'
```

Step 1 - install pods

```
$ pod install --verbose --no-repo-update
Preparing
Analyzing dependencies
...
Downloading dependencies
```

Podfile → podspec of each pod

```
AFNetworking.podspec
s.source = { :git => 'https://github.com/AFNetworking/AFNetworking.git'...}
s.public_header_files = 'AFNetworking/*.h'
s.source_files = 'AFNetworking/AFNetworking.h'

ss.source_files = 'AFNetworking/AFURL{Request,Response}Serialization.{h,m}'
ss.source_files = 'AFNetworking/AFSecurityPolicy.{h,m}'
ss.source_files = 'AFNetworking/AFNetworkReachabilityManager.{h,m}'
ss.source_files = 'AFNetworking/AFURLConnectionOperation.{h,m}',
                  'AFNetworking/AFHTTPRequestOperation.{h,m}',
                  'AFNetworking/AFHTTPRequestOperationManager.{h,m}'
ss.source_files = 'AFNetworking/AFURLSessionManager.{h,m}',
                  'AFNetworking/AFHTTPSessionManager.{h,m}'

ss.ios.public_header_files = 'UIKit+AFNetworking/*.h'
ss.ios.source_files = 'UIKit+AFNetworking' #ref a folder
```

```
-> Installing AFNetworking (2.5.4)
> Copying AFNetworking from `...Cache/.../AFNetworking/2.5.4-05edc` to `Pods/AFNetworking`
- Running pre install hooks
```

```
Generating Pods project
- Creating Pods project
- Adding source files to Pods project
- Adding frameworks to Pods project
- Adding libraries to Pods project
- Adding resources to Pods project
- Linking headers
- Installing targets
```

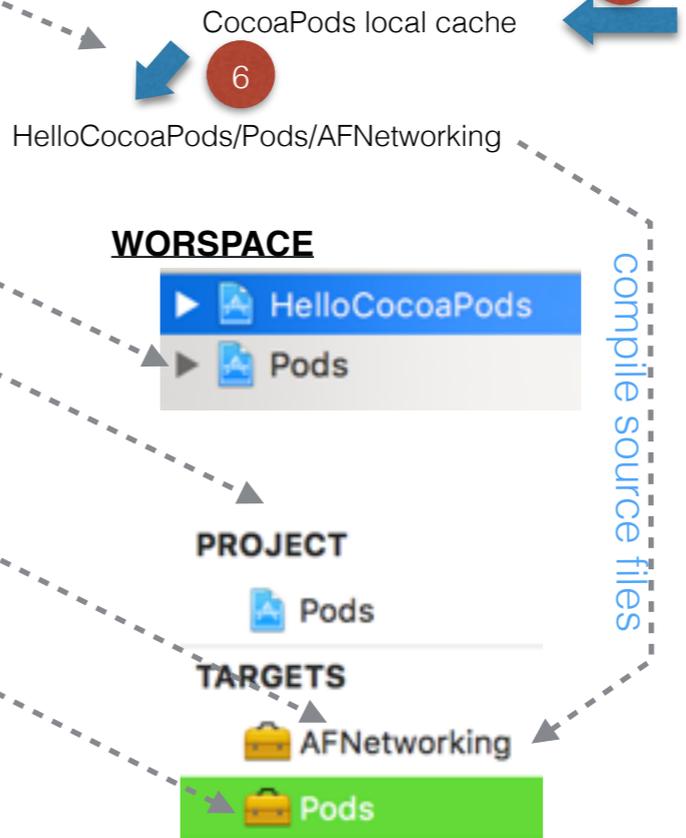
```
Installing target `AFNetworking` iOS 9.0
- Generating Info.plist file at `Pods/Target Support Files/AFNetworking/Info.plist`
- Generating module map file at `Pods/Target Support Files/AFNetworking/AFNetworking.modulemap`
- Generating umbrella header at `Pods/Target Support Files/AFNetworking/AFNetworking-umbrella.h`
```

```
Installing target `Pods` iOS 9.0
- Generating Info.plist file at `Pods/Target Support Files/Pods/Info.plist`
- Generating module map file at `Pods/Target Support Files/Pods/Pods.modulemap`
- Generating umbrella header at `Pods/Target Support Files/Pods/Pods-umbrella.h`
- Running post install hooks
- Writing Xcode project file to `Pods/Pods.xcodeproj`
- Writing Lockfile in `Podfile.lock`
- Writing Manifest in `Pods/Manifest.lock`
```

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

```
Integrating target `Pods` (`HelloCocoaPods.xcodeproj` project)
Adding Build Phase 'Embed Pods Frameworks' to project.
Adding Build Phase 'Copy Pods Resources' to project.
Adding Build Phase 'Check Pods Manifest.lock' to project.
- Running post install hooks
- cocoapods-stats from `.../cocoapods_plugin.rb`
```

```
Sending stats
- AFNetworking, 2.5.4
```



- AFNetworking
 - AFHTTPRequestOperation.h
 - AFHTTPRequestOperation.m
 - AFHTTPRequestOperationManager.h
 - AFHTTPRequestOperationManager.m
 - AFHTTPSessionManager.h
 - AFHTTPSessionManager.m
 - AFNetworkReachabilityManager.h
 - AFNetworkReachabilityManager.m
 - AFNetworking.h
 - AFSecurityPolicy.h
 - AFSecurityPolicy.m
 - AFURLConnectionOperation.h
 - AFURLConnectionOperation.m
 - AFURLRequestSerialization.h
 - AFURLRequestSerialization.m
 - AFURLResponseSerialization.h
 - AFURLResponseSerialization.m
 - AFURLSessionManager.h
 - AFURLSessionManager.m
- LICENSE
- README.md
- UIKit+AFNetworking
 - AFNetworkActivityIndicatorManager.h
 - AFNetworkActivityIndicatorManager.m
 - UIActivityIndicatorView+AFNetworking.h
 - UIActivityIndicatorView+AFNetworking.m
 - UIAlertView+AFNetworking.h
 - UIAlertView+AFNetworking.m
 - UIButton+AFNetworking.h
 - UIButton+AFNetworking.m
 - UIImageView+AFNetworking.h
 - UIImageView+AFNetworking.m
 - UIKit+AFNetworking.h
 - UIProgressView+AFNetworking.h
 - UIProgressView+AFNetworking.m
 - UIRefreshControl+AFNetworking.h
 - UIRefreshControl+AFNetworking.m
 - UIWebView+AFNetworking.h
 - UIWebView+AFNetworking.m

- user add Podfile to project, this sample syntax applies default target (MyPod.app) only
- user run 'pod install' in project
- Pods search source for pods, get and parse .podspec
- Pods get file list from .podspec
- cocoapods cache files
- Pods copy files into project
- Pods generate Pods.xcproject
- Pods create target for pod in Pods project
- Pods create Pods target in Pods project
- Pods create workspace, add projects in and update them

(Step 1) Targets & Build Config

PROJECT

HelloCocoaPods

TARGETS

- HelloCocoaPods** ▶ Target Dependencies (0 items)
- HelloCocoaPodsTests ▶ Compile Sources (3 items)
- HelloCocoaPodsUITests ▶ Link Binary With Libraries (0 items)
- ▶ Copy Bundle Resources (3 items)

Runpath Search Paths @executable_path/Frameworks

WORKSPACE

- HelloCocoaPods
- Pods

PROJECT

HelloCocoaPods

TARGETS

- HelloCocoaPods**
- HelloCocoaPodsTests
- HelloCocoaPodsUITests

PROJECT

Pods

TARGETS

- AFNetworking
- Pods**

Dep

- ▶ Target Dependencies (0 items)
- ▶ Check Pods Manifest.lock
- ▶ Compile Sources (3 items)
- ▶ Link Binary With Libraries (1 item)
- ▶ Copy Bundle Resources (3 items)
- ▶ Embed Pods Frameworks
- ▶ Copy Pods Resources

copy Pod.framework

copy compiled res of Pod.framework

Changed config

Runpath Search Paths '@executable_path/Frameworks' '@loader_path/Frameworks' @executable_path/Frameworks

Added config

Other Linker Flags -framework "AFNetworking"

Preprocessor Macros	<Multiple values>
Debug	DEBUG=1 COCOAPODS=1
Release	COCOAPODS=1
PODS_FRAMEWORK_BUILD_...	<Multiple values>
Debug	build/Debug-iphones/Pods
Release	build/Release-iphones/Pods
PODS_ROOT	/Users/pinxue/projects/study/ios_dev/...

Apple LLVM 7.0 - Custom Compiler Flags

Setting	HelloCocoaPods
Other C Flags	<Multiple values>
Debug	-iquote "build/Debug-iphones/AFNetworking.framework/Headers"
Release	-iquote "build/Release-iphones/AFNetworking.framework/Headers"
Other C++ Flags	<Multiple values>
Debug	-iquote "build/Debug-iphones/AFNetworking.framework/Headers"
Release	-iquote "build/Release-iphones/AFNetworking.framework/Headers"
Other Warning Flags	

-iquote Search dir only for header files requested with #include "file"; they are not searched for #include <file>, before all directories specified by -I and before the standard system directories.

(Step 1) Dependency, more

- pod 'name' without context
 - applies to default active target
- link_with '...Tests', '...UITests', ...
 - applies to all the targets in the list
- target 'name', :exclusive => true do
 - apply to this target only

Step 2 - Create a Pod

```
$ pod lib create --verbose MyPod
```

Cloning `https://github.com/CocoaPods/pod-template.git` into `MyPod`.
\$ /usr/bin/git clone https://github.com/CocoaPods/pod-template.git MyPod
Cloning into 'MyPod'...

Configuring MyPod template.

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

[ObjC, No Demo, No Test Frameworks, Class Prefix=My]

Running pod install on your new library.

Updating local specs repositories

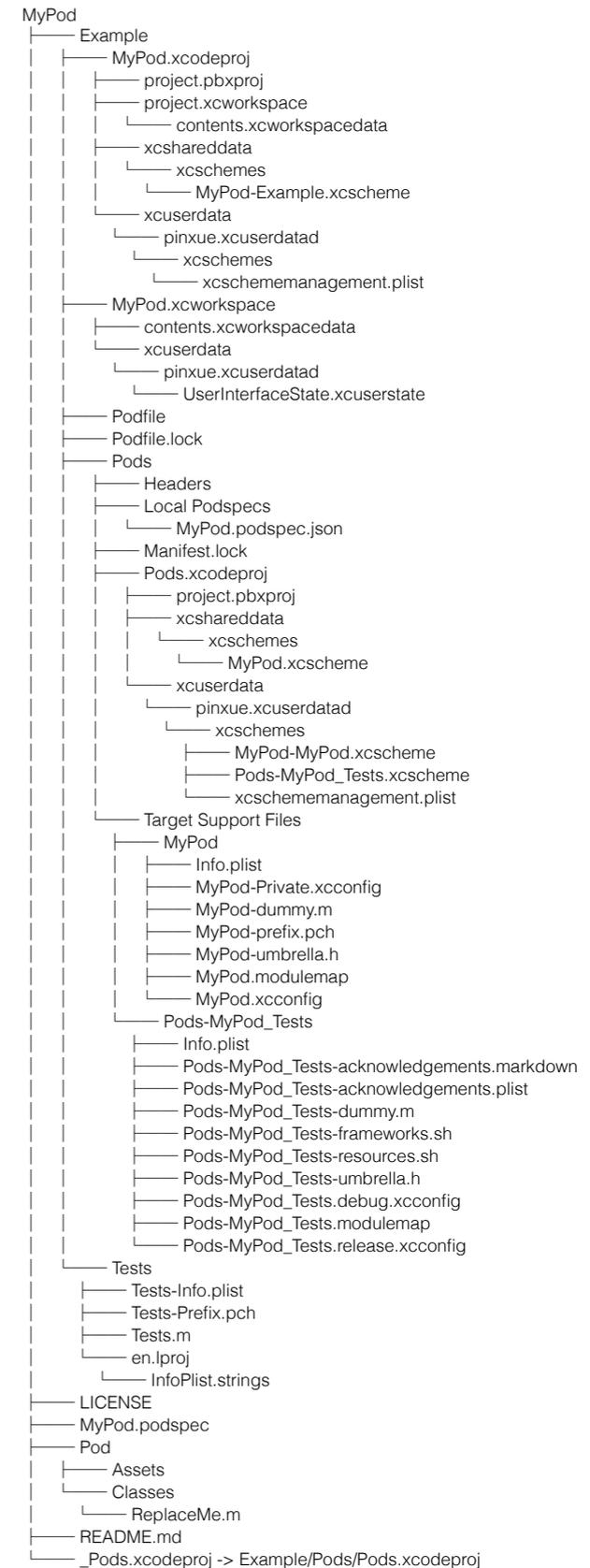
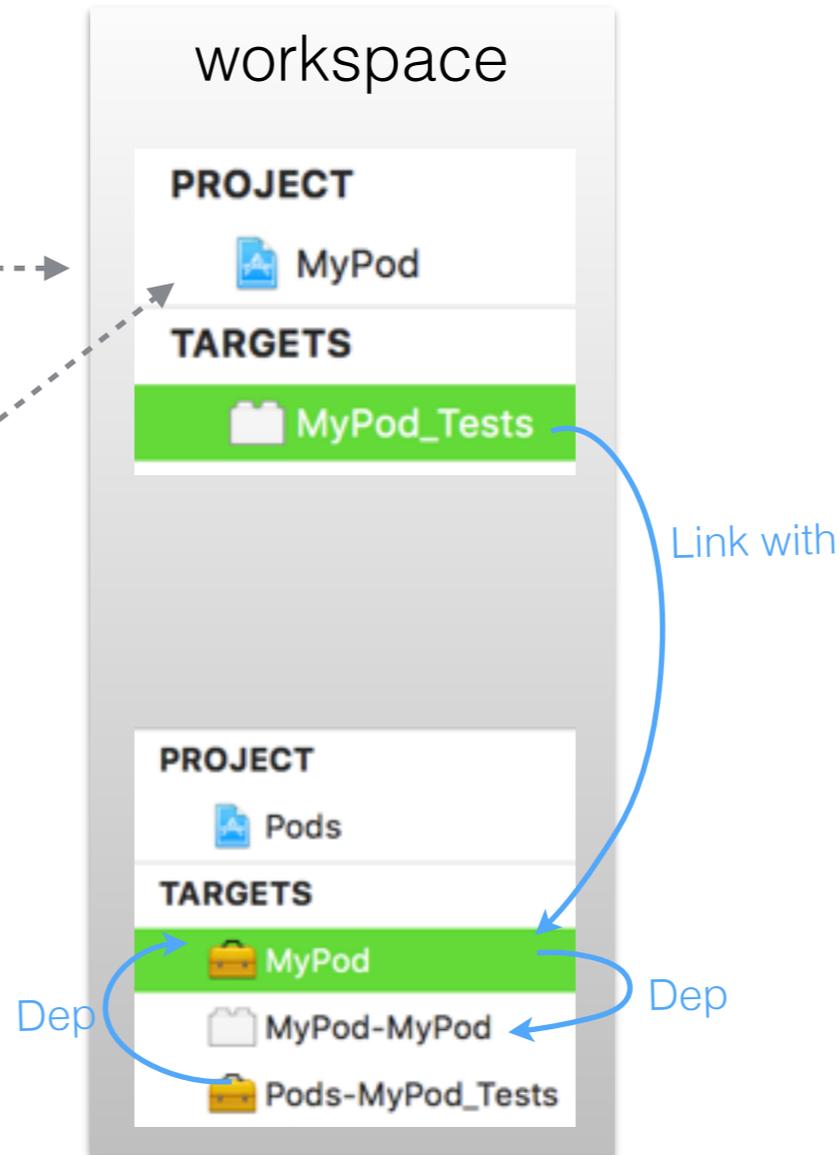
Analyzing dependencies
Fetching podspec for `MyPod` from `../`
Downloading dependencies
Installing MyPod (0.1.0)
Generating Pods project
Integrating client project

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

Sending stats

Ace! you're ready to go!
We will start you off by opening your project in Xcode
open 'MyPod/Example/MyPod.xcworkspace'

reference:
<http://blog.cocoapods.org/Pod-Lib-Create/>
<http://guides.cocoapods.org/making/using-pod-lib-create.html>



Step 3 - Add a dependency in .podspec

#no dependency

PROJECT

MyPod

TARGETS

MyPod_Example

MyPod_Tests

PROJECT

Pods

TARGETS

MyPod

MyPod-MyPod

Pods-MyPod_Example

Pods-MyPod_Tests

Link with

Dep

Dep

Dep

s.dependency 'AFNetworking', '~> 2.3'

PROJECT

MyPod

TARGETS

MyPod_Example

MyPod_Tests

PROJECT

Pods

TARGETS

AFNetworking

MyPod

MyPod-MyPod

Pods-MyPod_Example

Pods-MyPod_Tests

Link with

Dep

Dep

Link with

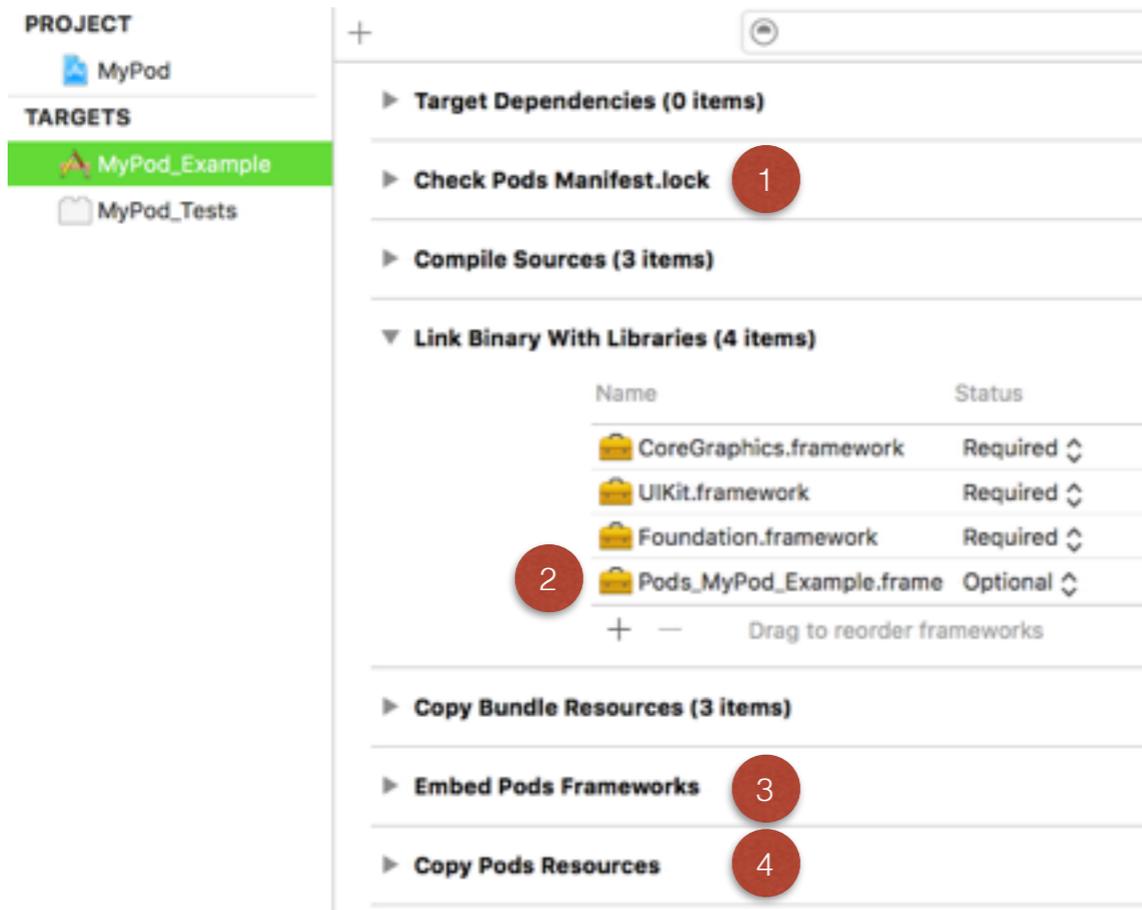
Dep

Dep

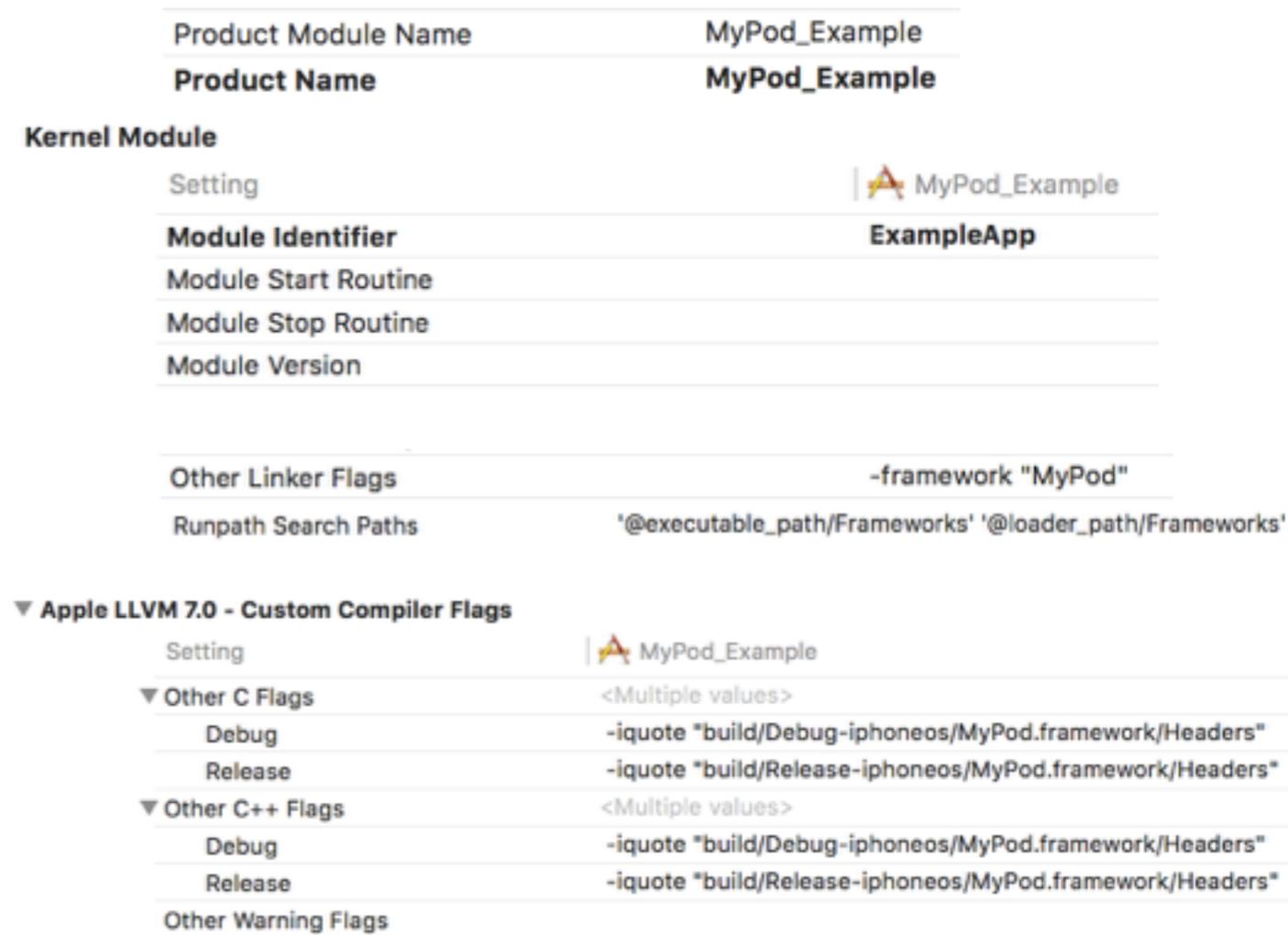
Dep

Dep

(Step 3) Example app: Target, Build Config

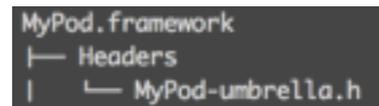


1. check consistent of /Podfile.lock and Pods/manifest.lock
2. link to development pod
3. copy pod framework into example.app
4. copy pod resource into example.app

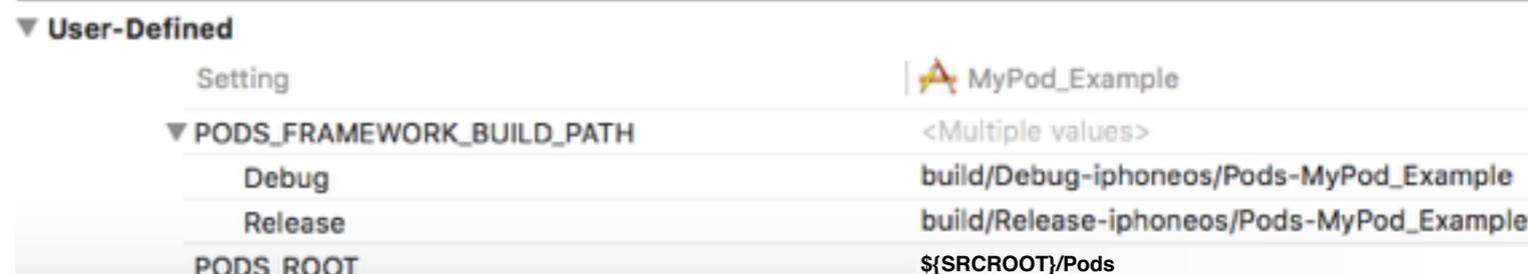


-iquote Search *dir* only for header files requested with #include "file"; they are not searched for #include <file>, before all directories specified by -I and before the standard system directories.

implicit included by all source files



Prefix Header MyPod/MyPod-Prefix.pch

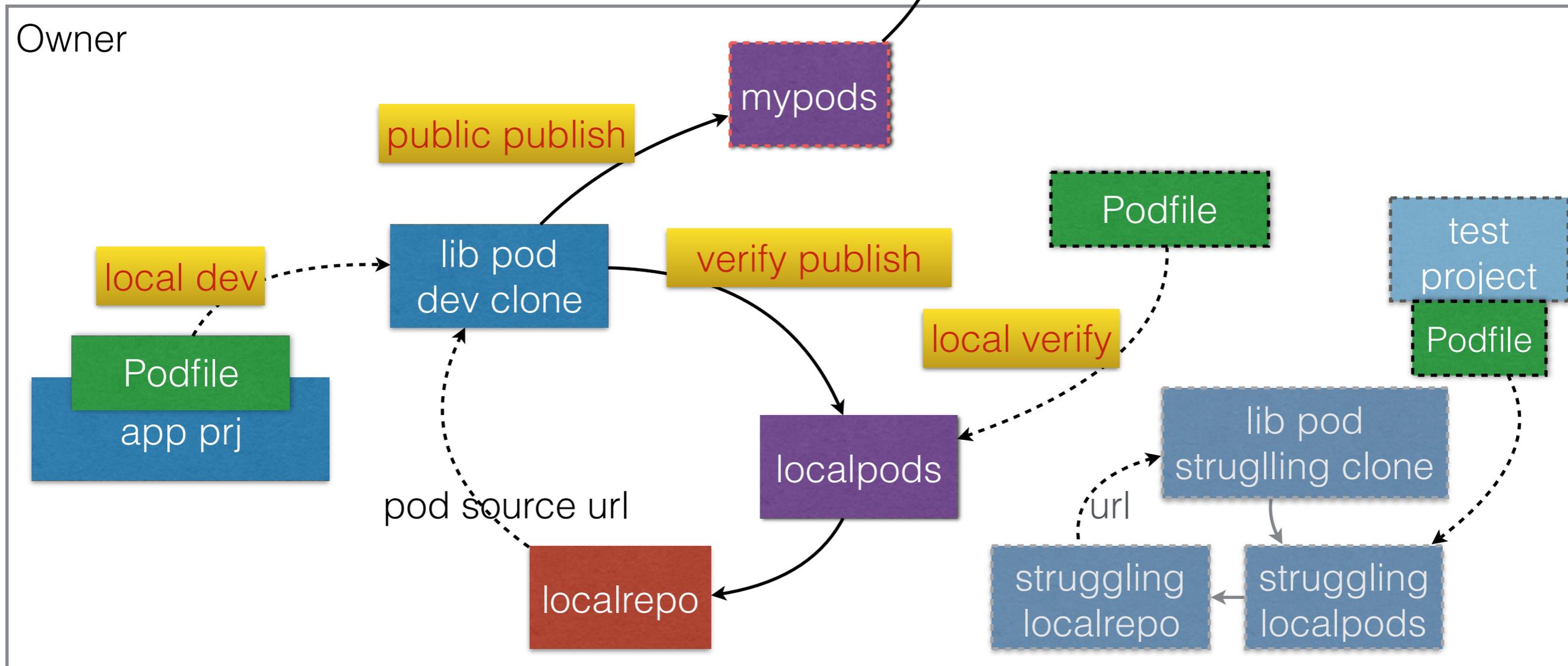
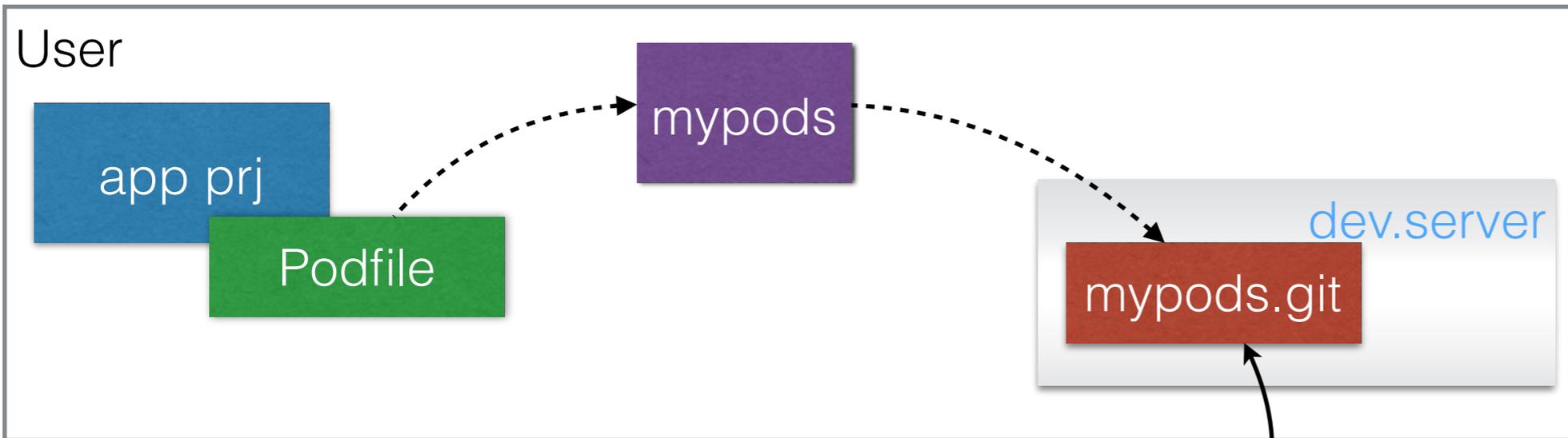


Practice

- add CocoaPods into an existed project
- start a new project with CocoaPods from beginning
- create a pod with example
- use the pod created locally
- restructure a project to pods

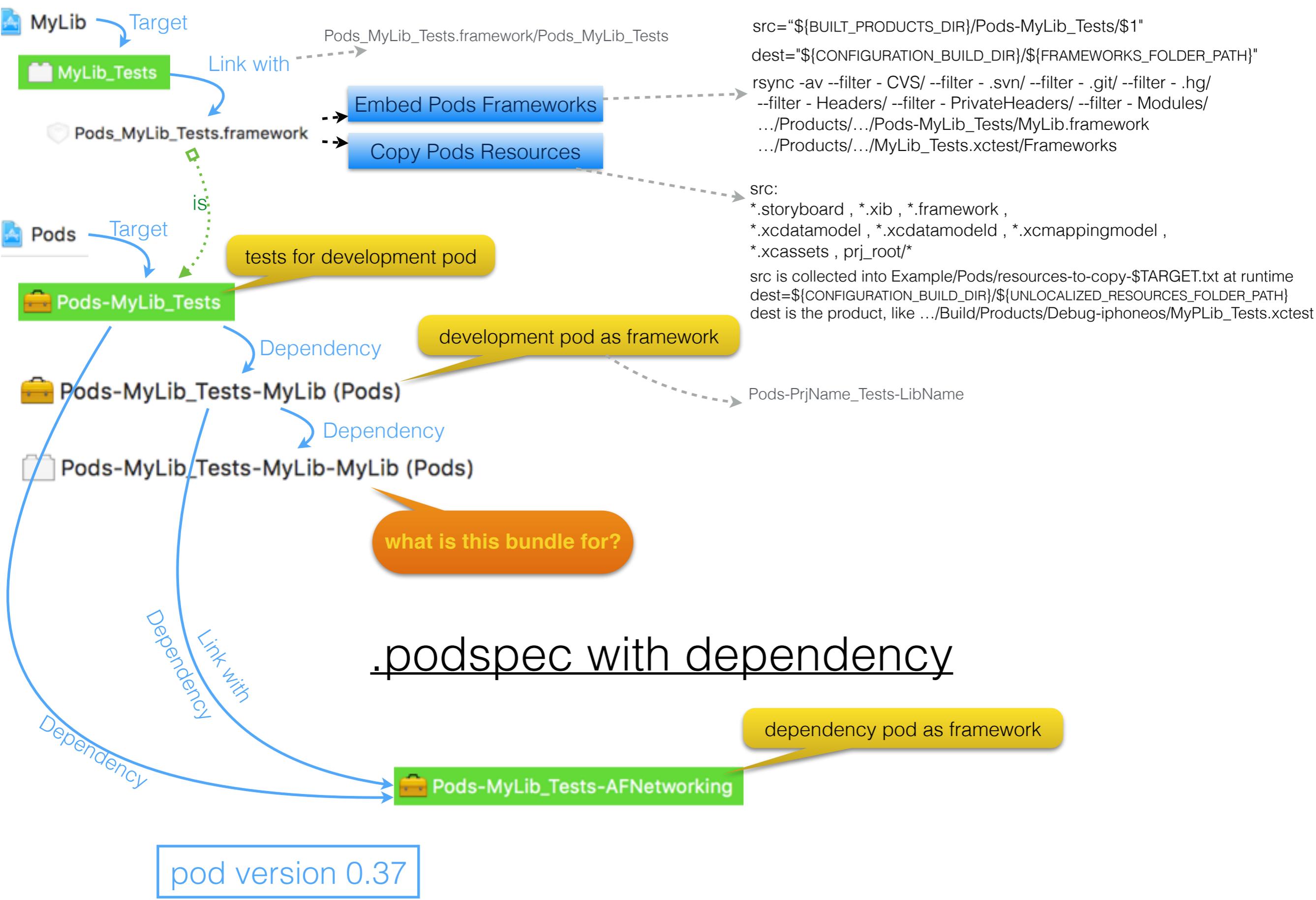
Setup local repo

- Setup
 - `git init --bare local_spec_repo`
 - `pod repo add path/to/local_spec_repo local`
 - `pod repo list`
- Use
 - `pod lib lint -- sources='local' myLib.podspec`
 - `pod repo push local myLib.podspec`
 - Podfile

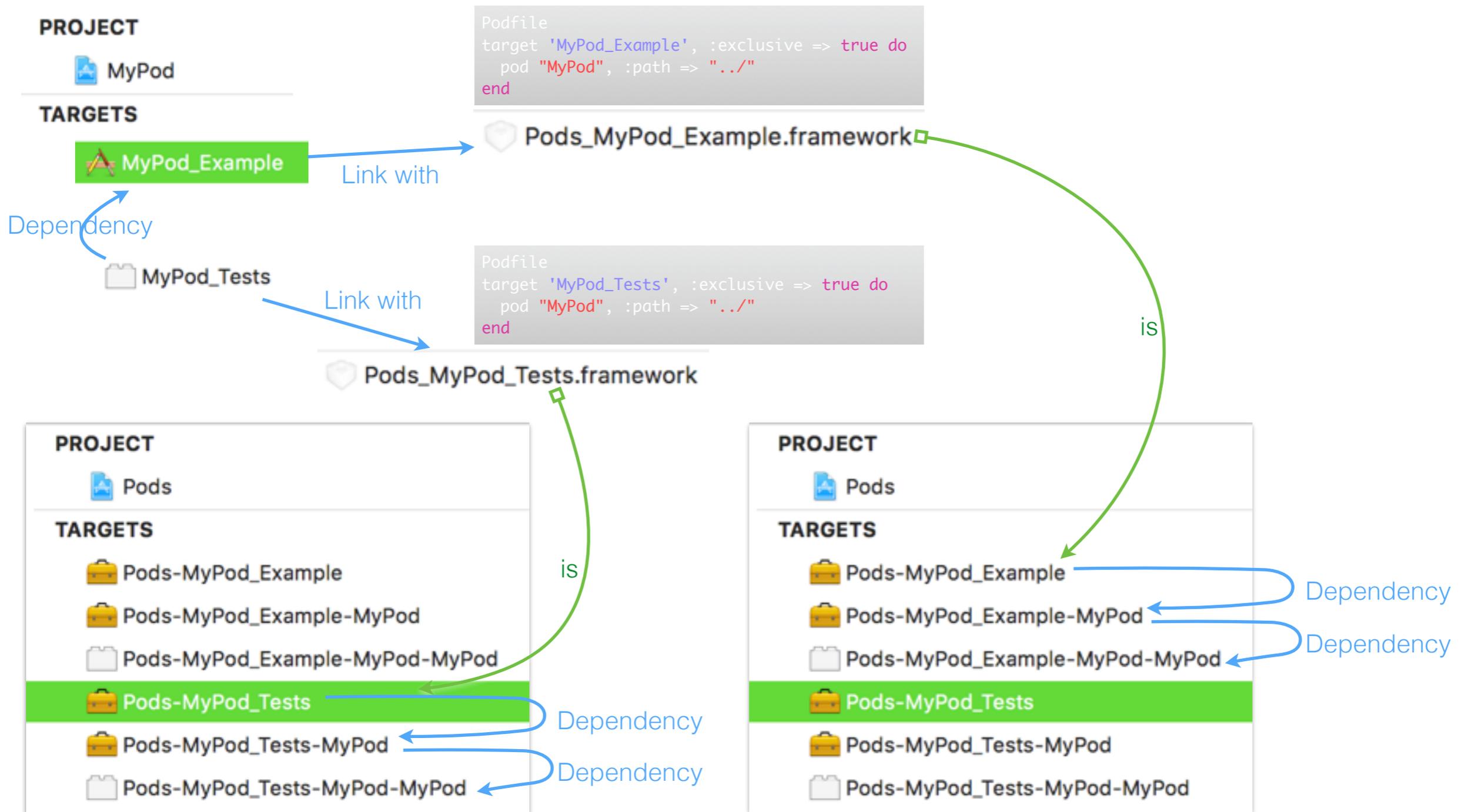


Impl in 0.3.7

Targets of Basic CocoaPods pod project

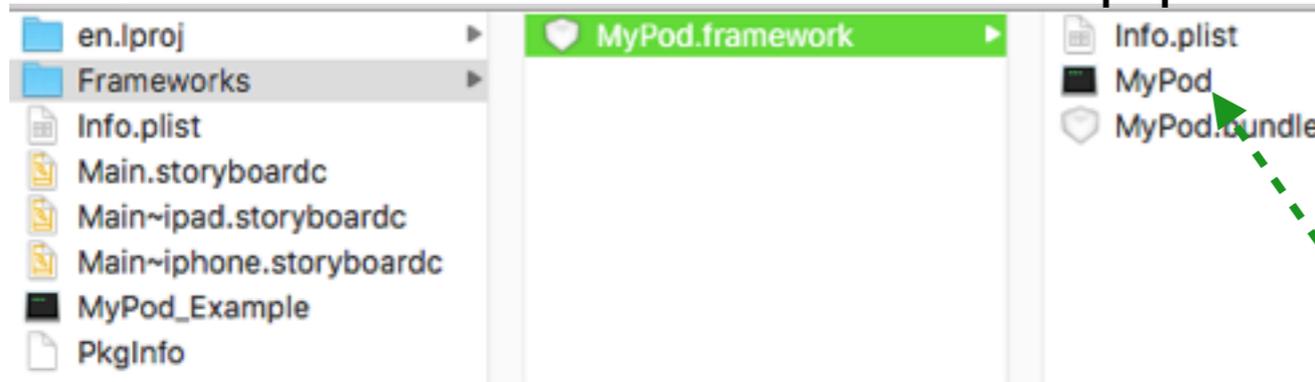


Targets of pod lib created with demo app

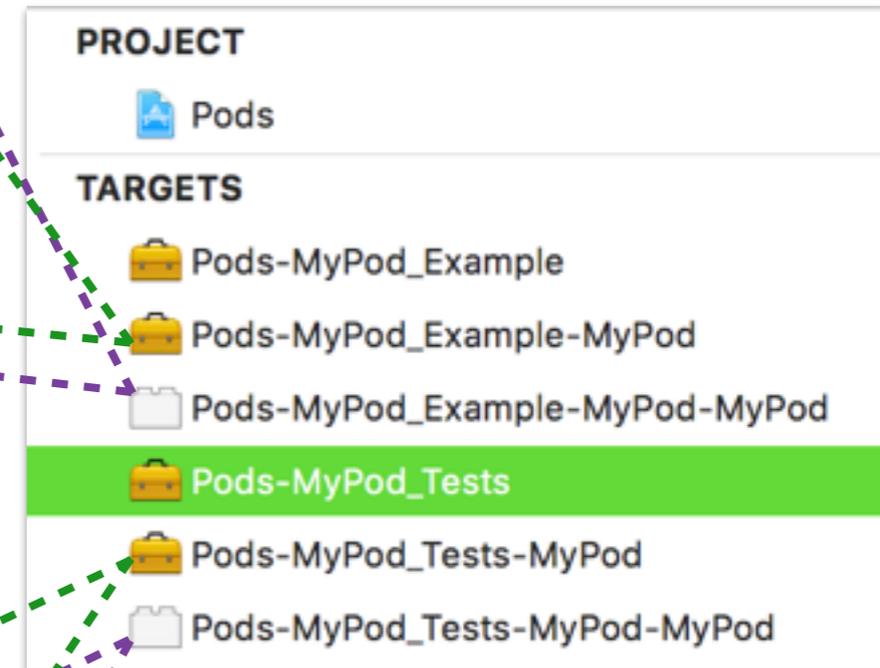
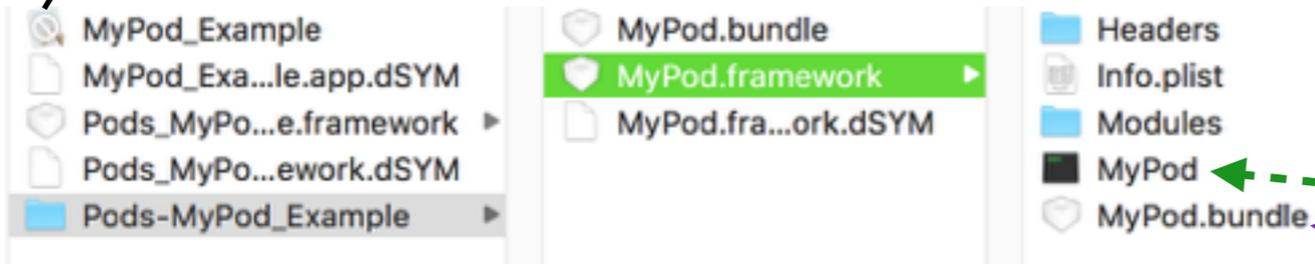


pod version 0.37

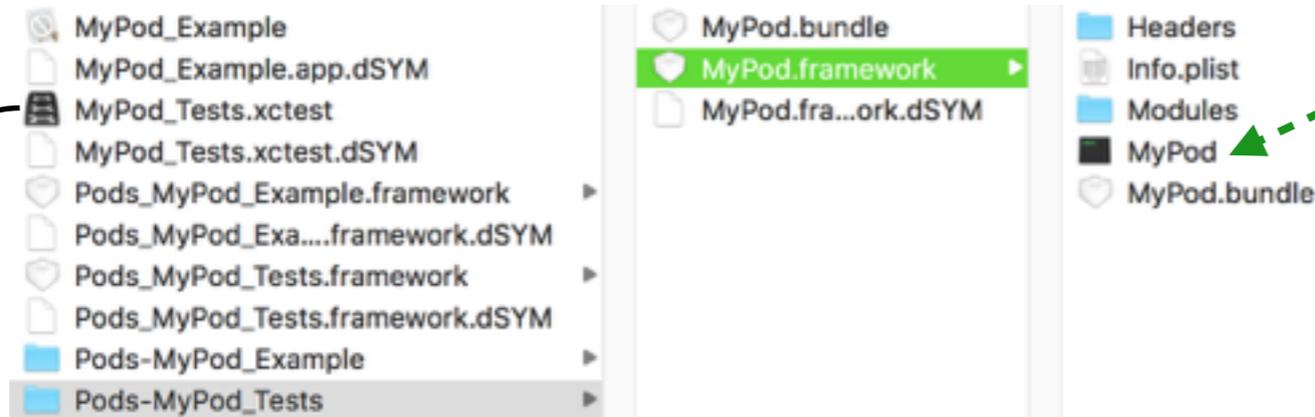
Embed framework in app



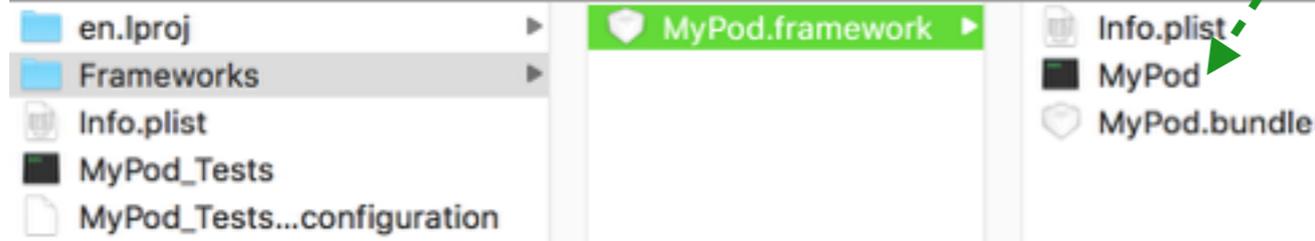
Built products for app



Built products for test



Embed framework in test



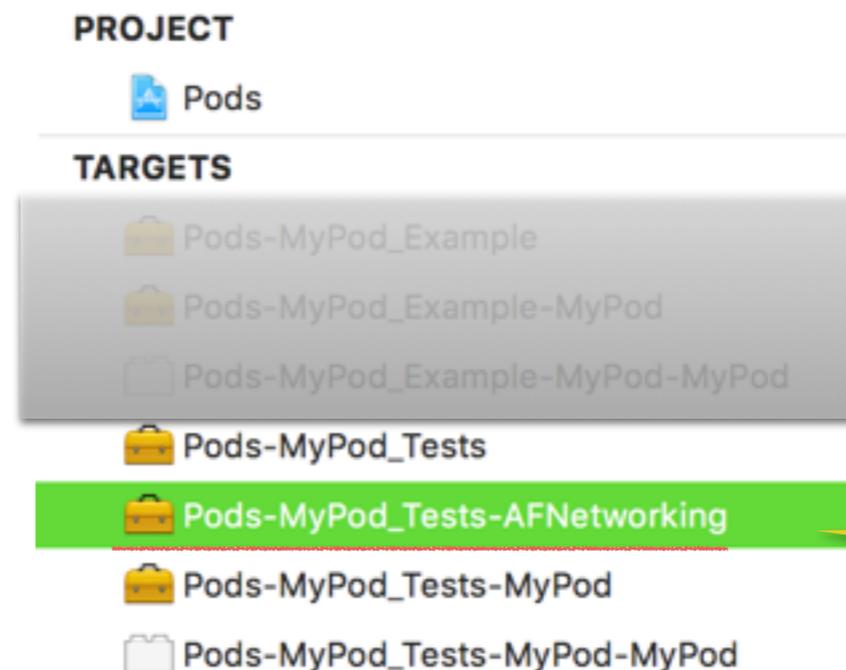
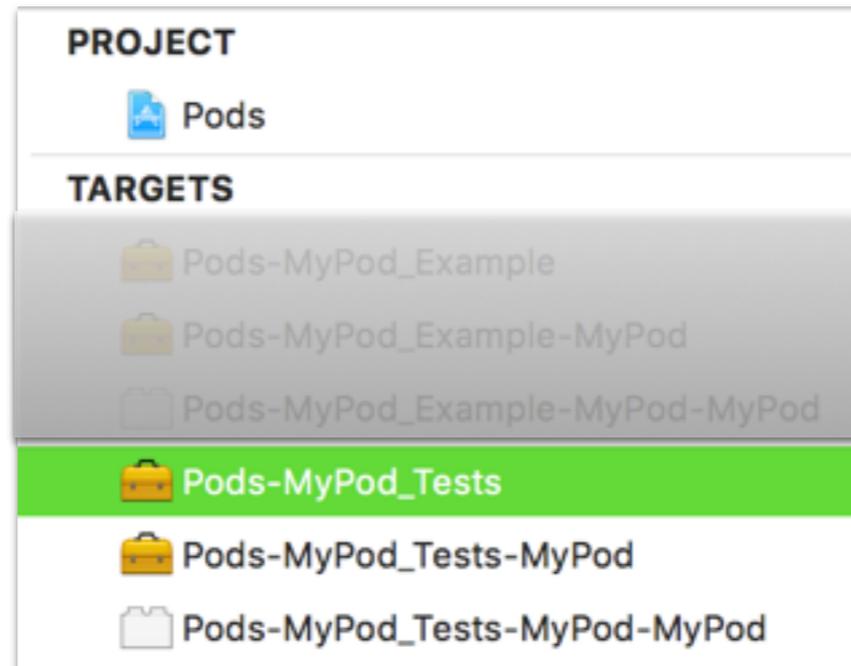
pod version 0.37

Just add a dependency in Podfile for test target

```
.podspec  
target 'MyPod_Tests', :exclusive => true do  
  pod "MyPod", :path => "../"  
end
```



```
.podspec  
target 'MyPod_Tests', :exclusive => true do  
  pod "MyPod", :path => "../"  
  pod "AFNetworking"  
end
```



build of dev-pod for demo-app doesn't use AFNetworking, AFNetworking will NOT appear in MyPod_Example.app

dev-pod test depends on AFNetworking. will appear in MyPod_Tests.xctest