Publishing Your First CocoaPod

Pietro Rea
iOS Engineer @ Upside Travel

Getting started

- CocoaPods made its debut in 2011. It’s the mainstream package manager in 2018.
- Other package managers also available. Carthage and SwiftPM are the main ones.
- **Everyone** uses it but **few** publish to it.

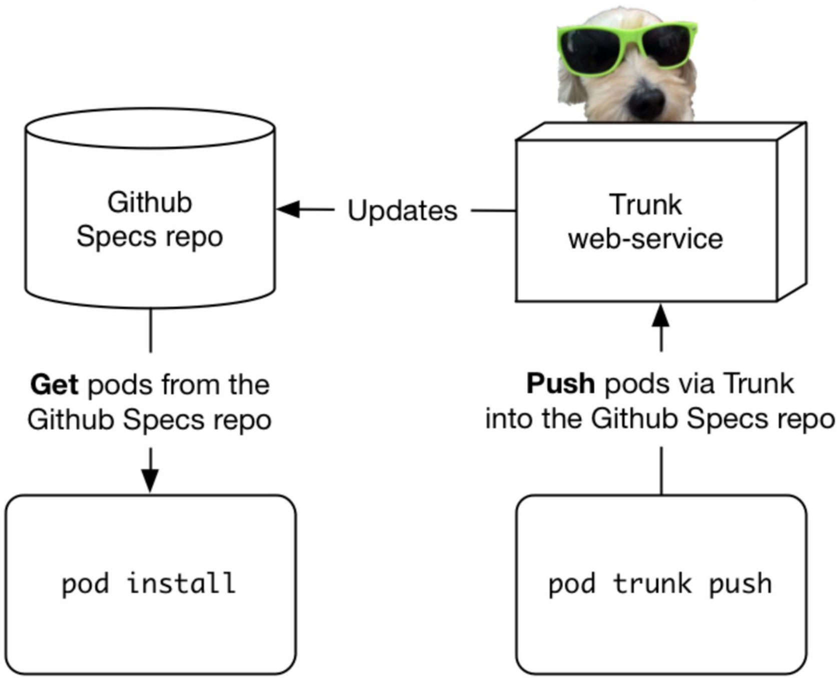
The Mechanics

0. Installing Cocoapods
1. Getting set up with Trunk.
2. Create a new pod from the default template.
3. Add your code to the pod.
4. Lint your podspec & test your pod
5. Publish your pod
6. …
7. Profit!

What is CocoaPods Trunk?

- Problem: Only the “owner” of a pod should update it. How do we establish “ownership”?
- Answer: Authentication required. Enter **Trunk**.
- Trunk is an authentication API service for Cocoapods.
- Trunk doesn’t use passwords. Ties a session to your machine.

What is CocoaPods Trunk?



Github Specs repo

Updates

Trunk web-service

Get pods from the Github Specs repo

Push pods via Trunk into the Github Specs repo

`pod install`

`pod trunk push`

Getting setup with Trunk

-
  ```
  $ pod trunk register your@emailaddress.com 'Your Name' --description='Session name'
  ```

- Confirmation link sent to your email address. Click to register yourself and start a session.

- Verify your current session: `$ pod trunk me`

Using the default pod template

```
$ pod lib create SpringAnimationController
```

- “What platform do you want to use?” [**iOS** / macOS]
- “What language do you want to use?” [**Swift** / ObjC]
- “Would you like to include a demo application with your library?” [**Yes** / No]
- “Which testing frameworks will you use?” [**Quick** / None]
- “Would you like to do view based testing?” [**Yes** / No]

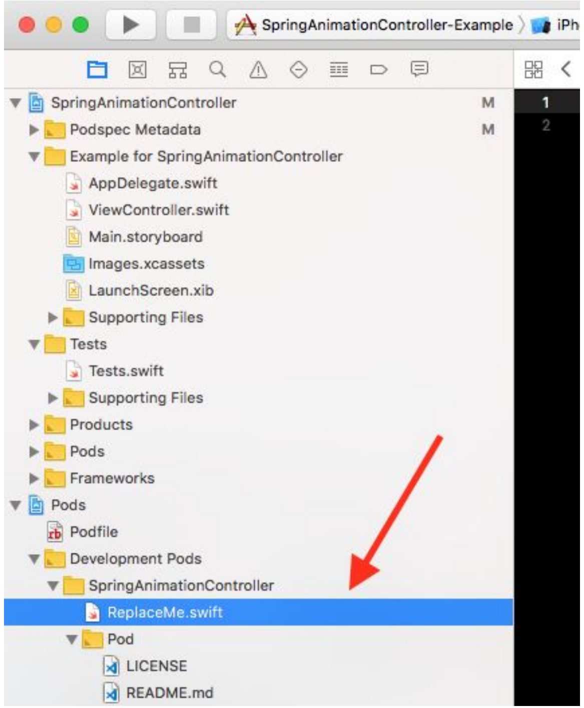
Change the podspec

```
Pod::Spec.new do |s|
  s.name             = 'SpringAnimationController'
  s.version          = '0.1.0'
  s.summary          = 'A short description of SpringAnimationController. 140 characters only!'
  s.description      = 'Add your description here'
  s.homepage         = 'https://github.com/pietrorea/SpringAnimationController'
  # s.screenshots     = 'www.example.com/screenshots_1', 'www.example.com/screenshots_2'
  s.license          = { :type => 'MIT', :file => 'LICENSE' }
  s.author           = { 'pietrorea' => 'your@email.com' }
  s.source           = { :git => 'https://github.com/pietrorea/SpringAnimationController.git', :tag =>
s.version.to_s }
  # s.social_media_url = 'https://twitter.com/pietrorea'

  s.ios.deployment_target = '8.0'
  s.source_files = 'SpringAnimationController/Classes/**/*'
end
```

Move your code into the pod

- **Two** targets: one for your demo project and one for your pod.
- CocoaPods looks at the file pattern in **s.source_files** to find your code.
- Beware of **access control** problems. Entities have to be **public** to be visible to host app!



Testing/linting your pod

- Test before publishing to with trunk
    - Change podspec to use **branch** instead of **tag**
    - Push changes to GitHub
    - `pod ‘SpringAnimationController’, :git => ..., :branch => branchName` in Podfile
    - `$ pod install`
- Lint your podspec before publishing
    - `$ pod spec lint`
    - Linting catches basic problems (e.g configuration)

Publishing your work

-
  ```
  $ git tag -a 1.0.0 -m ‘1.0.0` && git push --tags
  ```
  - Add a tag* when you’re done making changes to your pod (i.e. pod source files, sample project, README, podspect)

-
  ```
  $ pod trunk push
  SpringAnimationController.podspec
  ```

-
  ```
  $ pod try SpringAnimationController
  ```
  - Use `pod try` to verify you can pull from CocoaPods.

Celebrate!

```
----------------------------------------------------------------------------
🎉  Congrats

🚀  SpringAnimationController (1.0.0) successfully published
📅  June 11th, 02:16
□  https://cocoapods.org/pods/SpringAnimationController
👍  Tell your friends!
----------------------------------------------------------------------------
```

Okay, but what do I publish??

- **Idea #1**: Pull out something reusable from a side project. E.g. SpringAnimationController

- **Idea #2**: Your very own bag ‘o tricks. E.g. SSToolkit

- **Idea #3**: Keep track of **schleps** you have to solve over and over. Some common ones: configuring `NSAttributedString`, writing out Auto Layout constraints manually, etc.

Judge me by my lines of code, do you?

- **Failable v 0.1.0:** https://cocoapods.org/pods/Failable
- **30,844** total downloads.
- 34 lines of code (but really 4)

```
public enum Failable<T> {
    case success(T)
    case failure(Error)
}
```

Embrace the letters Y and A

- **Ya**hoo! (Yet another hierarchical, officious oracle!)
- **YA**ML (Yet another markup language)
- **Ya**ndex (Yet another indexer)
- **YA**CC (Yet another compiler compiler)
- https://en.wikipedia.org/wiki/Yet_another
- [Insert your CocoaPod here]

Questions?

Pietro Rea
iOS Engineer @ Upside Travel