SD Hacks iOS Workshop Setup Guide

Hey hacker! Thanks for showing interest in this iOS workshop! We're super glad you showed up, and we can't wait to you in person! However, before the workshop actually happens, we'd like to set up some parts of the project beforehand to make everybody's experience as smooth as possible.

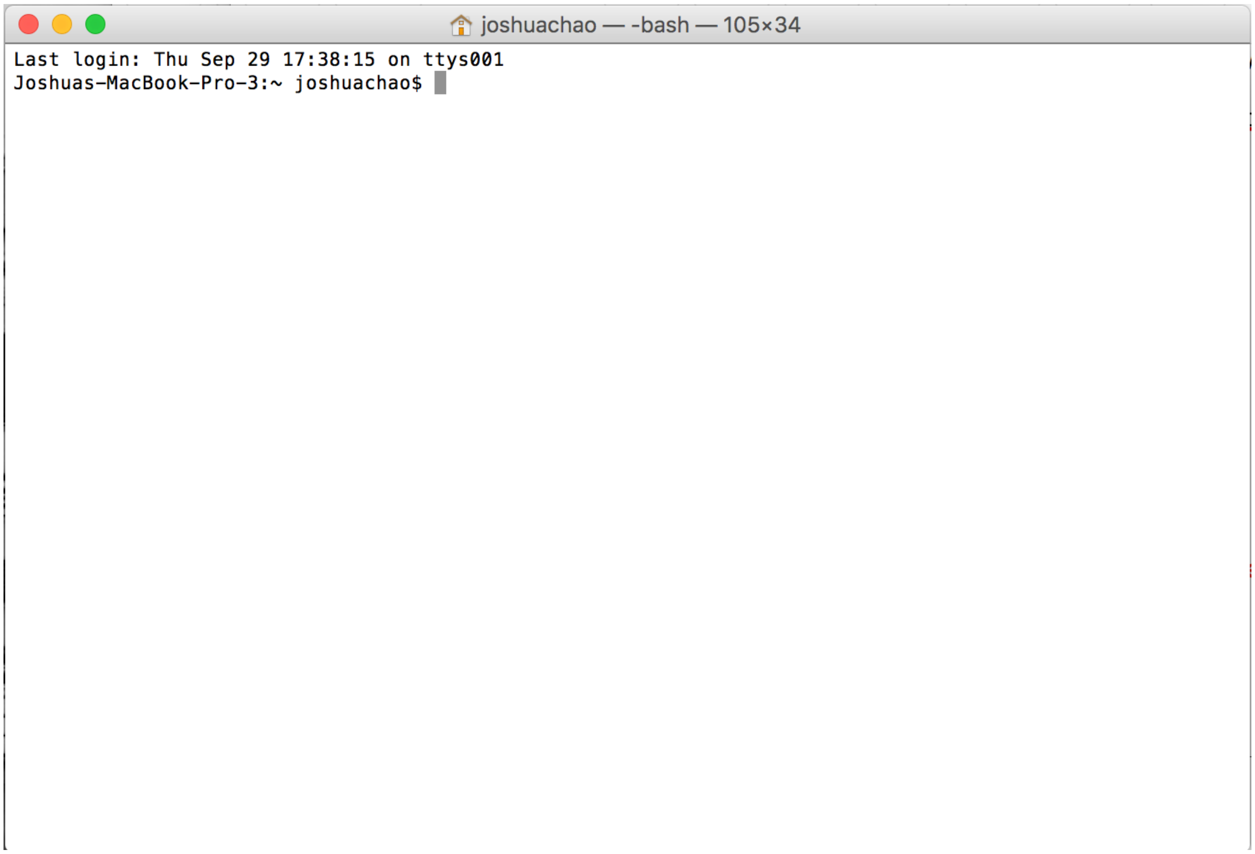
We're going to make a Twitter clone using Firebase today! No worries if you don't know what Firebase is; we're going to be going over that extensively at the workshop. However, we are going to have to download some stuff here and there so we can use Firebase.

System Requirements

1. **Have a Mac.** Sorry, we're not going to be running over Windows in this workshop. If you DO have a Windows and don't have a Mac, make friends and buddy up with a partner who does! It's called pair programming, and it actually helps you code better, faster, and understand what you're typing more clearly!
2. **Make sure to download XCode!** It doesn't matter if you don't know what XCode is, but you can find XCode on the App Store, by typing "XCode" into the search bar. The icon looks like this:

Make sure to download XCode 8! **If you already have XCode, make sure to update to XCode 8.** This may or may not require you to update your computer OS, and downloading XCode itself might take a while, so sit back, grab a soda, and relax.

3. **Install XCode Developer Tools.** This allows you to do some necessary stuff that normal people don't really need, so just for safety's sake, Apple requires you to install it manually. Open up a Terminal! To open up Terminal, type in Command + Space, type in "Terminal", and then press "Enter". Something like this should pop up:



```
Last login: Thu Sep 29 17:38:15 on ttys001
Joshuas-MacBook-Pro-3:~ joshuachao$ 
```

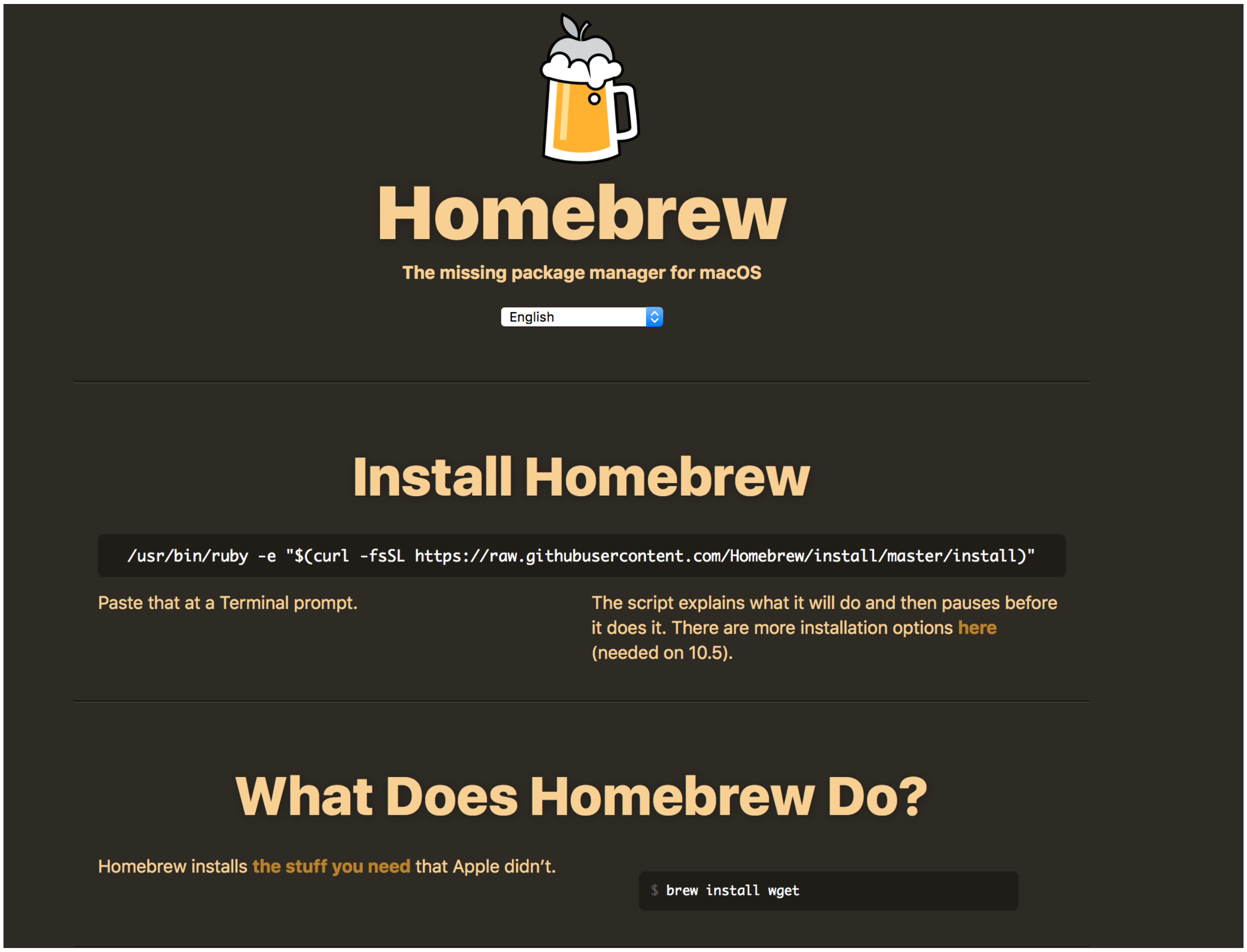
This might look scary, but don't worry! It actually makes you look really cool, I promise. In the Terminal, type in:

```
xcode-select --install
```

Follow the instructions to install XCode command line tools.

4. **Time to download Homebrew!** Homebrew is a really handy dandy package manager for Mac, but again, it's totally okay if you don't know what any of those words mean. Go ahead and visit this website: http://brew.sh/index.html

   It should look something like this:



Homebrew

The missing package manager for macOS

Install Homebrew

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Paste that at a Terminal prompt.

The script explains what it will do and then pauses before it does it. There are more installation options here (needed on 10.5).

What Does Homebrew Do?

Homebrew installs the stuff you need that Apple didn't.

```
$ brew install wget
```

Now, copy the long line Homebrew kindly tells you to copy, and paste it into your Terminal. Press “Enter”. Now, follow the steps that pop up on the Terminal in order to install Homebrew. **If you already have Homebrew installed, go ahead and type:**

```
brew update
```

If you can’t remember whether or not you’ve already installed Homebrew, type the above line anyways; Homebrew will tell you whether or not you’ve installed Homebrew before. Nifty, huh?

5. We’re now gonna use Homebrew to **install and upgrade Ruby!** Ruby is a really handy language that can talk to websites really well. In a later step, we’re going to install Cocoapods, and that runs on top of Ruby. Again, you don’t need to know any of this stuff. Back in the Terminal, type in:

   ```
   brew install ruby
   ```

   Press “Enter”. Most likely, your cursor will move to the next line and do absolutely nothing. **Don’t type anything into the Terminal at this point.** Terminal does this frustrating thing where it won’t tell you if it’s running anything or not, but trust me, it’s running. After a while, a whole bunch of text will pop up on the screen, showing the download process of installing Ruby.

   After all that finishes, go ahead and type in:

   ```
   brew upgrade ruby
   ```

Once again, Terminal will "hang", but don't freak out! Homebrew's just doing it's stuff.

6. We're now going to use Ruby to **download Cocoapods**! Cocoapods is an iOS dependency manager. Don't worry if you don't know what that means. All it does is allow your iOS app to use Firebase properly. The Cocoapods website can be found at: https://cocoapods.org

   Go ahead and type in:

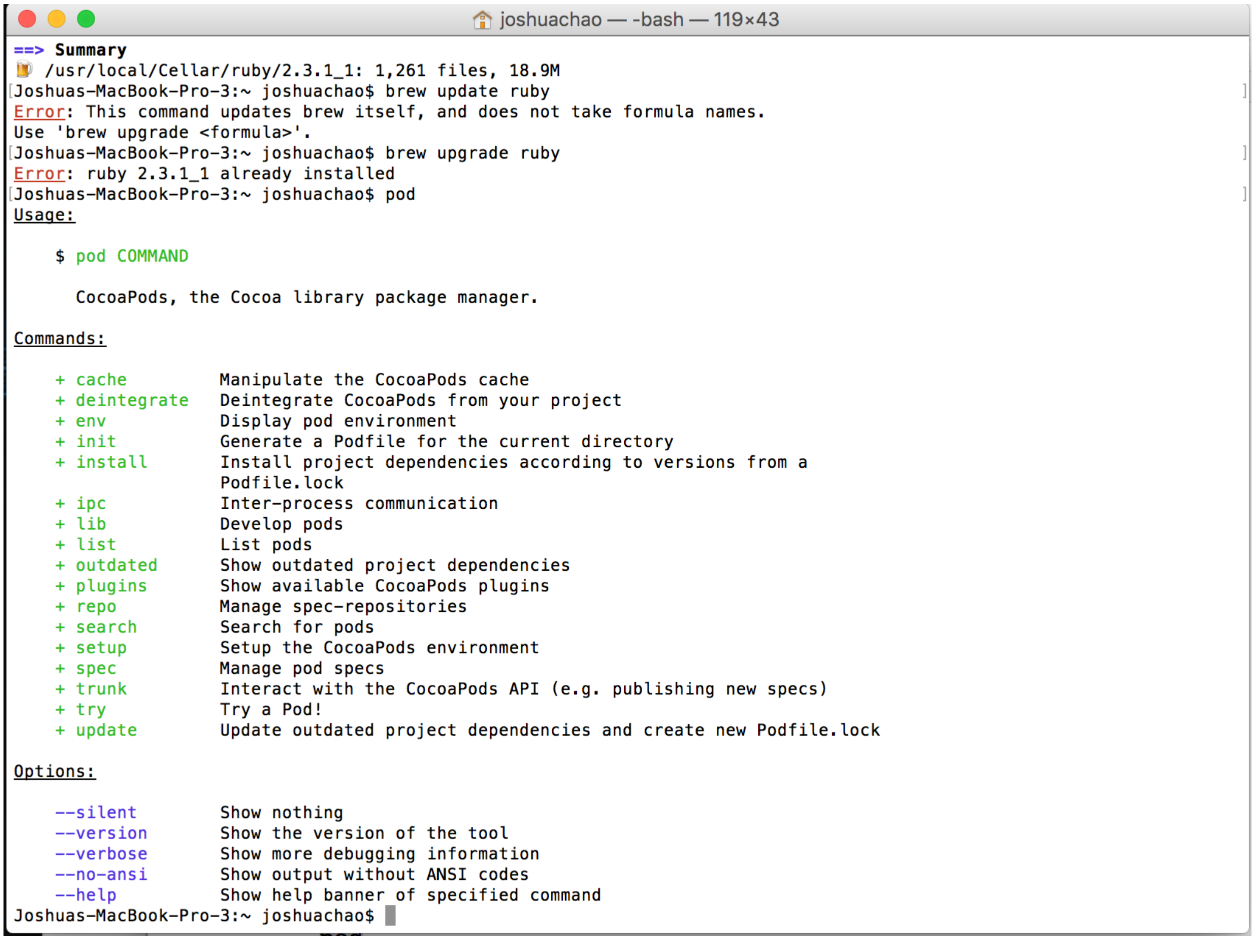
   ```
   sudo gem install cocoapods
   ```

   into the Terminal, and press "Enter". Again, Terminal will pause and pretend to do nothing, but just kick back and chill with your fellow hackers.

   At this point, Terminal will ask you to type in your password. When you do, **nothing will pop up on the screen**. This is Terminal's way of maintaining password secrecy. You're just going to have a leap of faith, type in your password through muscle memory, and press "Enter" when you're done. If you get lost, just press Backspace a bunch of times until you're sure everything that you've typed is cleared.

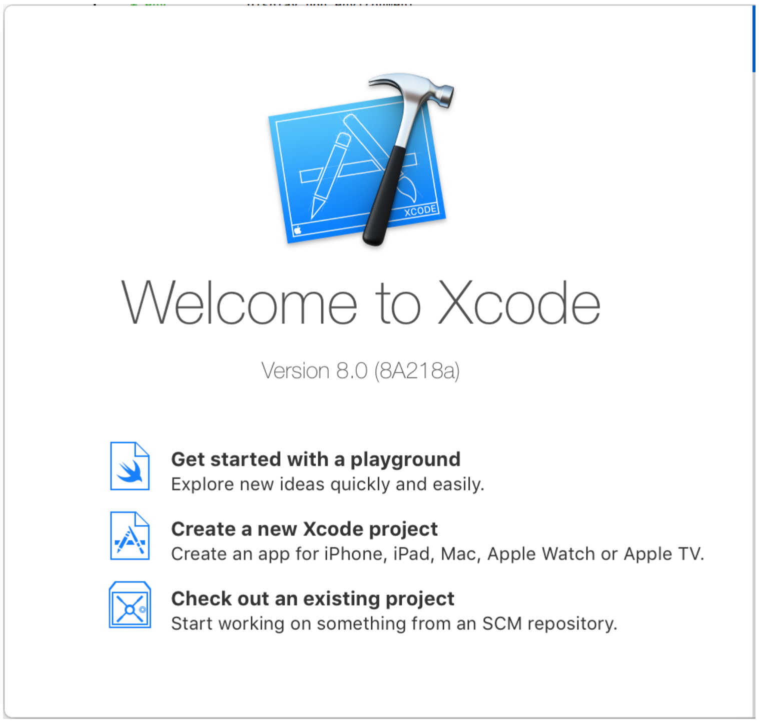
   Once Cocoapods has finished installing, you can verify everything is good by typing in:

   ```
   pod
   ```



```
==> Summary
🍺  /usr/local/Cellar/ruby/2.3.1_1: 1,261 files, 18.9M
Joshuas-MacBook-Pro-3:~ joshuachao$ brew update ruby
Error: This command updates brew itself, and does not take formula names.
Use 'brew upgrade <formula>'.
Joshuas-MacBook-Pro-3:~ joshuachao$ brew upgrade ruby
Error: ruby 2.3.1_1 already installed
Joshuas-MacBook-Pro-3:~ joshuachao$ pod
Usage:

    $ pod COMMAND

      CocoaPods, the Cocoa library package manager.

Commands:

    + cache         Manipulate the CocoaPods cache
    + deintegrate   Deintegrate CocoaPods from your project
    + env           Display pod environment
    + init          Generate a Podfile for the current directory
    + install       Install project dependencies according to versions from a
                    Podfile.lock
    + ipc           Inter-process communication
    + lib           Develop pods
    + list          List pods
    + outdated      Show outdated project dependencies
    + plugins       Show available CocoaPods plugins
    + repo          Manage spec-repositories
    + search        Search for pods
    + setup         Setup the CocoaPods environment
    + spec          Manage pod specs
    + trunk         Interact with the CocoaPods API (e.g. publishing new specs)
    + try           Try a Pod!
    + update        Update outdated project dependencies and create new Podfile.lock

Options:

    --silent        Show nothing
    --version       Show the version of the tool
    --verbose       Show more debugging information
    --no-ansi       Show output without ANSI codes
    --help          Show help banner of specified command
Joshuas-MacBook-Pro-3:~ joshuachao$ 
```

7. Let's go ahead and set up our project! Go ahead and open up XCode, and something like this should pop up:



Welcome to Xcode

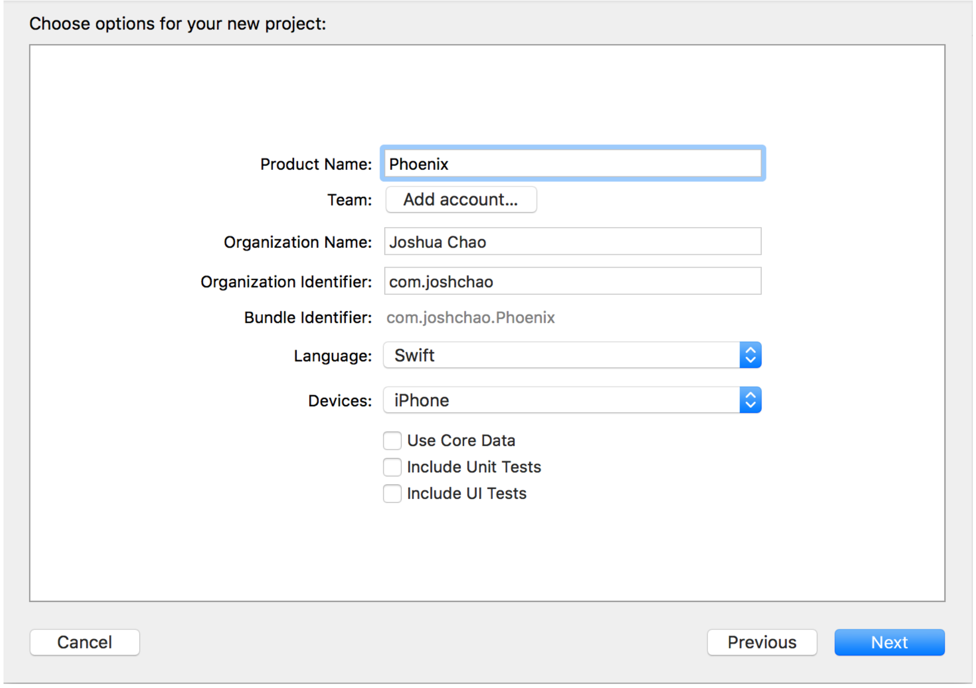
Version 8.0 (8A218a)

Get started with a playground
Explore new ideas quickly and easily.

Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.

Check out an existing project
Start working on something from an SCM repository.

8. Click on “**Create a new XCode Project**”, select “**Single View Application**”, and for the product name, type in “Phoenix” (Twitter = bird, Firebase = flame, bird + flame = phoenix lol). For the organization name, put in your own name, and for the organization identifier, type something that can be completely unique. Convention states you start with “com.” + unique tag name you’re comfortable with. It can be your Github username, your gamer tag, whatever you’d like! Choose “**Swift**” as your language, choose “iPhone” as your device, and completely uncheck the three options (Core Data, Unit Tests, and UI Tests). When you’re done, it should look something like this:



Choose options for your new project:

Product Name: Phoenix

Team: Add account...

Organization Name: Joshua Chao

Organization Identifier: com.joshchao

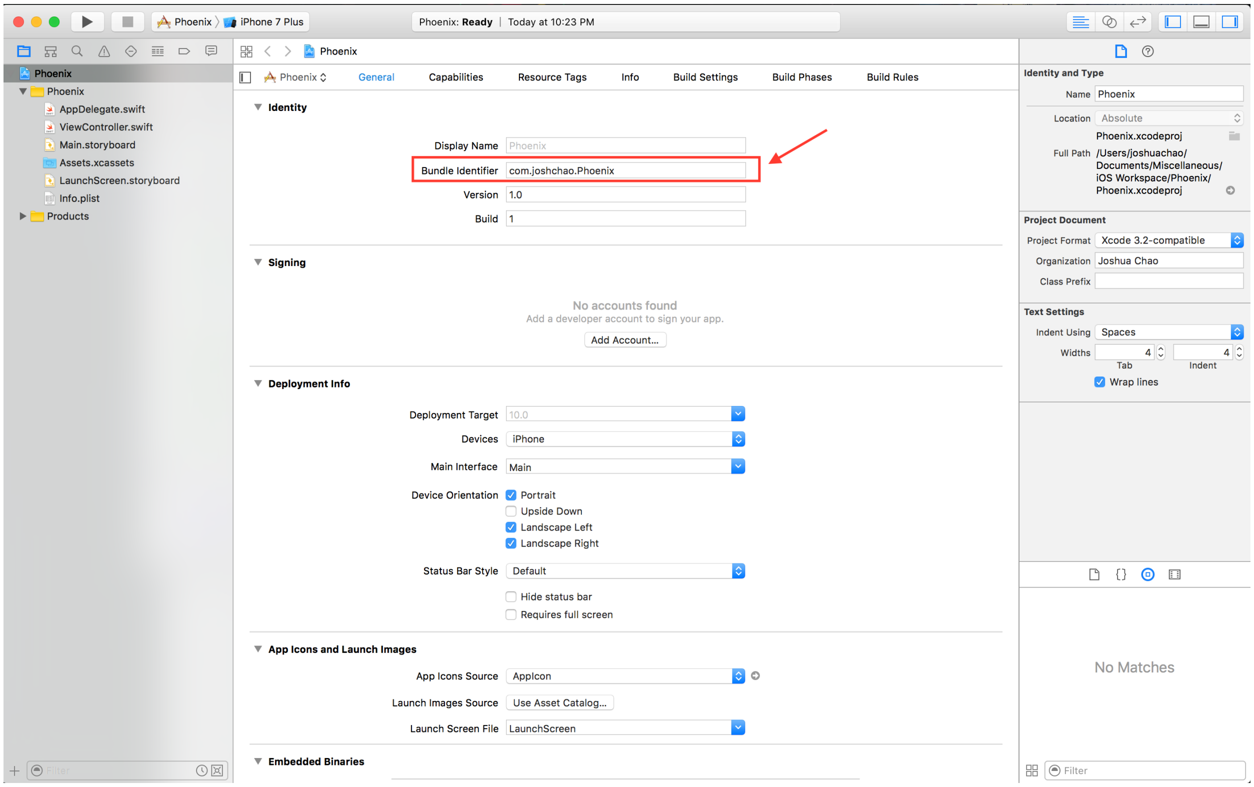
Bundle Identifier: com.joshchao.Phoenix

Language: Swift

Devices: iPhone

☐ Use Core Data
☐ Include Unit Tests
☐ Include UI Tests

9. Save your project wherever you'd like, but make sure you can find it later! When you're all finished, XCode should open something like this!



Identity

Display Name: Phoenix

Bundle Identifier: com.joshchao.Phoenix

Version: 1.0

Build: 1

Signing

No accounts found
Add a developer account to sign your app.

Deployment Info

Deployment Target: 10.0

Devices: iPhone

Main Interface: Main

Device Orientation: ☑ Portrait, ☐ Upside Down, ☑ Landscape Left, ☑ Landscape Right

Status Bar Style: Default

☐ Hide status bar

☐ Requires full screen

App Icons and Launch Images

App Icons Source: AppIcon

Launch Images Source: Use Asset Catalog...

Launch Screen File: LaunchScreen

Embedded Binaries

Identity and Type

Name: Phoenix

Location: Absolute

Phoenix.xcodeproj

Full Path: /Users/joshuachao/Documents/Miscellaneous/iOS Workspace/Phoenix/Phoenix.xcodeproj

Project Document

Project Format: Xcode 3.2-compatible

Organization: Joshua Chao

Class Prefix:

Text Settings

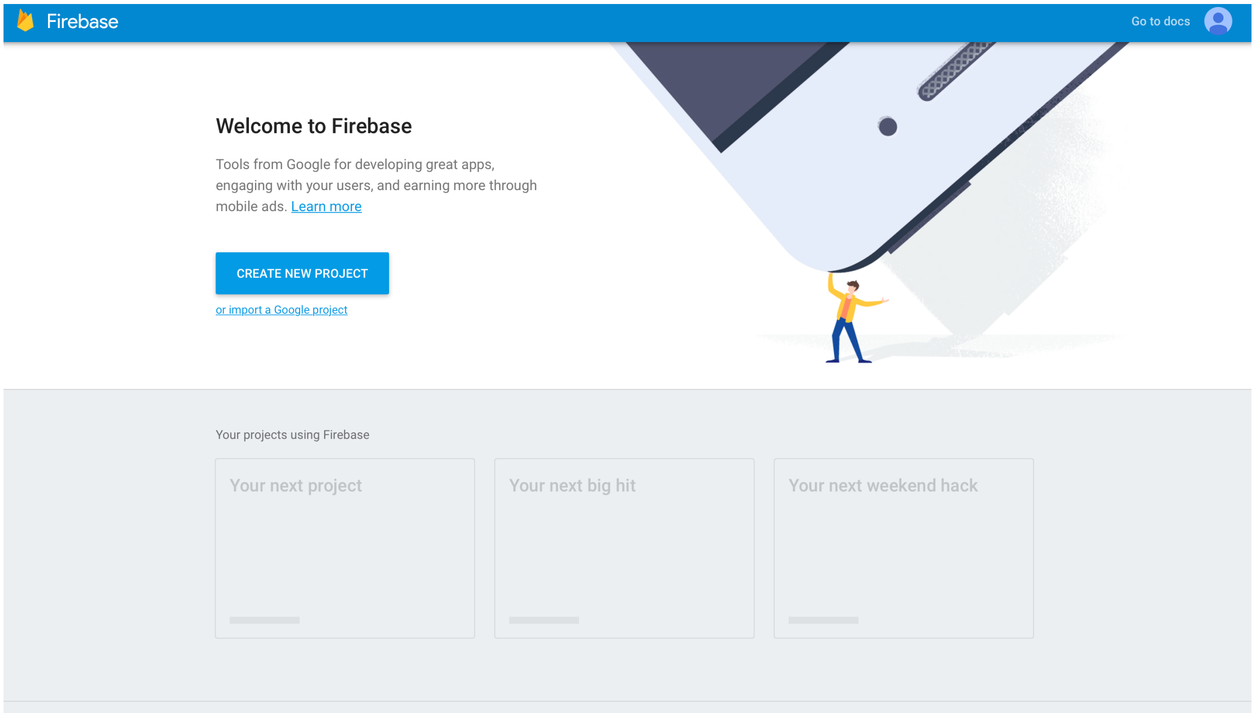
Indent Using: Spaces

Widths: 4 Tab, 4 Indent

☑ Wrap lines

At this point in time, make sure to **copy** the *bundle identifier* of your program (see above).

10. Time for some fun stuff now (not that I'm saying what we've been doing earlier hasn't been fun)! Go to firebase.google.com, click on "Go to Console in the top right link", and log into your Google account. Click on "Create a New Project" in the resulting page.



Firebase

Go to docs

Welcome to Firebase

Tools from Google for developing great apps, engaging with your users, and earning more through mobile ads. Learn more

CREATE NEW PROJECT

or import a Google project

Your projects using Firebase

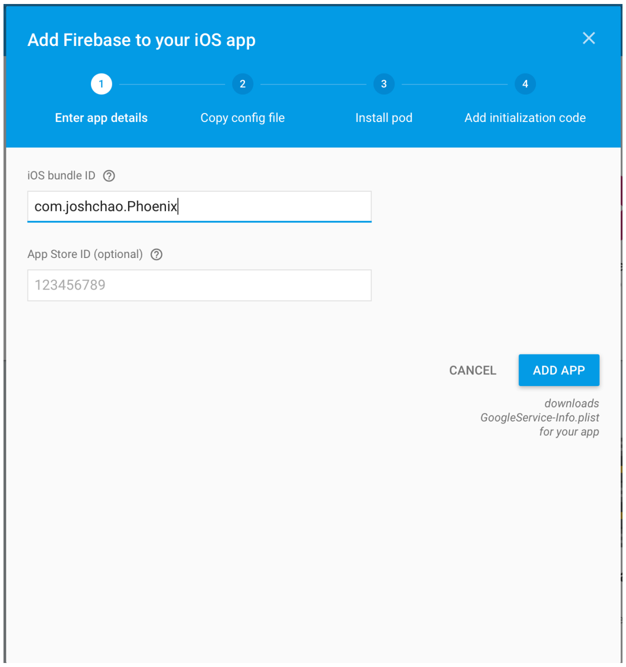
Your next project

Your next big hit

Your next weekend hack

Follow the step and input "Phoenix" as the project name.

11. Click on "Add Firebase to your iOS App". Now, in the resulting section, in the *iOS Bundle ID*, **paste the bundle identifier that you copied earlier.**



Add Firebase to your iOS app

1 Enter app details

2 Copy config file

3 Install pod

4 Add initialization code

iOS bundle ID: com.joshchao.Phoenix

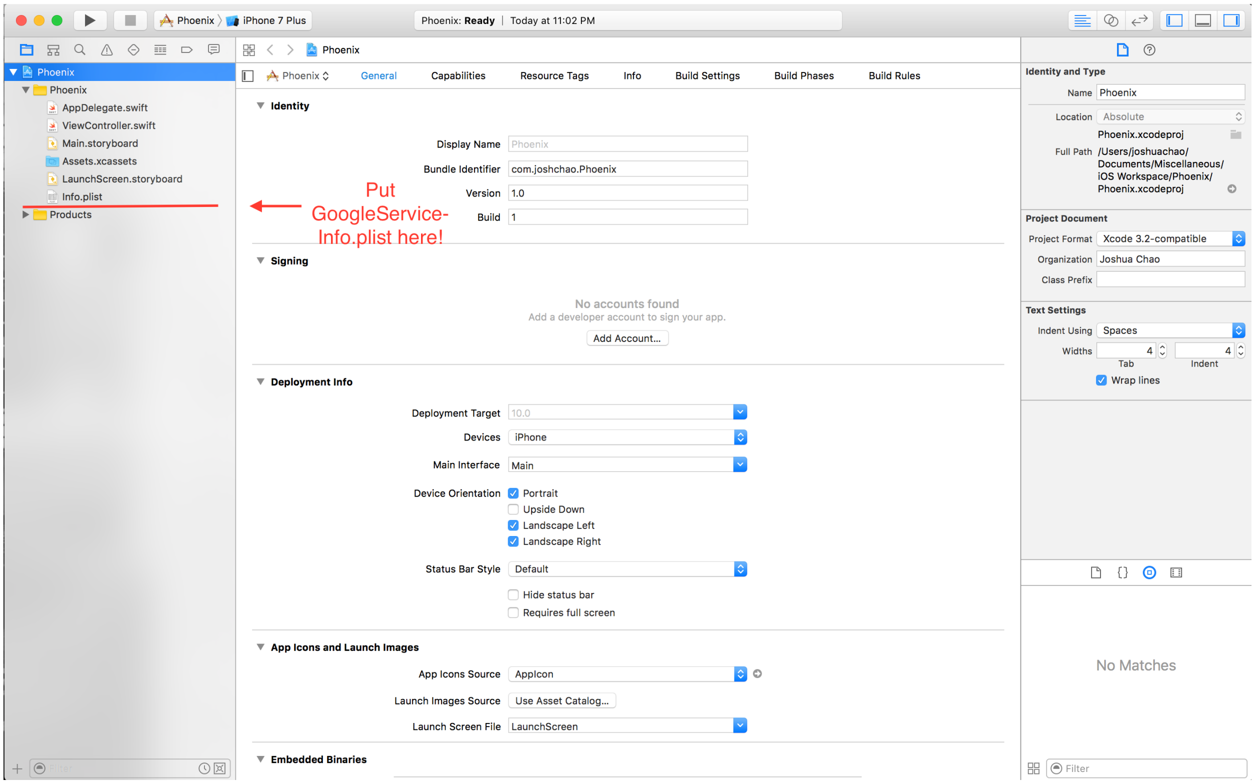
App Store ID (optional): 123456789

CANCEL

ADD APP

downloads GoogleService-Info.plist for your app

12. Click on “Add App”, and an instant download called **GoogleService-Info.plist** should be slapped into your Downloads folder. Now, go back to your XCode. **Drag GoogleServiceInfo.plist from your Downloads folder right underneath “Info.plist”** (a blue bar should pop up where you want it to be).



Put GoogleService-Info.plist here!

Identity

Display Name: Phoenix
Bundle Identifier: com.joshchao.Phoenix
Version: 1.0
Build: 1

Signing

No accounts found
Add a developer account to sign your app.

Deployment Info

Deployment Target: 10.0
Devices: iPhone
Main Interface: Main
Device Orientation: ☑ Portrait ☐ Upside Down ☑ Landscape Left ☑ Landscape Right
Status Bar Style: Default
☐ Hide status bar
☐ Requires full screen

App Icons and Launch Images

App Icons Source: AppIcon
Launch Images Source: Use Asset Catalog...
Launch Screen File: LaunchScreen

Embedded Binaries

Identity and Type

Name: Phoenix
Location: Absolute
Phoenix.xcodeproj
Full Path: /Users/joshuachao/Documents/Miscellaneous/iOS Workspace/Phoenix/Phoenix.xcodeproj

Project Document

Project Format: Xcode 3.2-compatible
Organization: Joshua Chao
Class Prefix:

Text Settings

Indent Using: Spaces
Widths: 4 Tab, 4 Indent
☑ Wrap lines

13. Now, go back into your Terminal, and we’re gonna learn some really basic Unix commands! For those of you who already know how to navigate around Terminal, go ahead and navigate your way into where you saved your XCode project.

You’re only gonna to learn three commands to become a Unix navigation god: **cd**, **ls**, and **pwd.** Your terminal can essentially be viewed as a text-based file-finder. Just like in the Apple Finder, where you can click on folders and jump in and out of them to view and open your files, you can do the same by typing in words (known as “commands”).

cd: This command allows you to actually jump in and out of folders. Use cd in the following way:

```
cd <NAME_OF_THE_FOLDER_YOU_WANT_TO_JUMP_INTO_HERE>
```

Note the space in between the folder name and the “cd” command. A special case of the cd command is “cd ..” (Note the space and two dots). This allows you to “jump back”, or hit the metaphorical back button to jump back where you came from.)

ls: This allows you to list all the stuff in the folder so that you can see what folders you can jump into or files you can open. When in doubt, always ls!

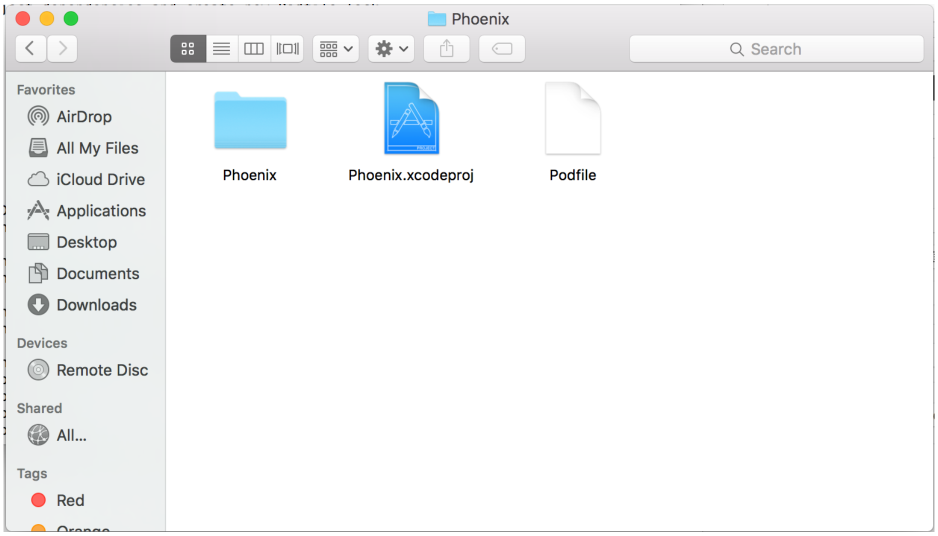
pwd: This will tell you where you are in case you get lost.

You’re gonna want to cd and ls until you jump into your Phoenix folder. If you ls, you should see “Phoenix” and “Phoenix.workspace” listed as the options.

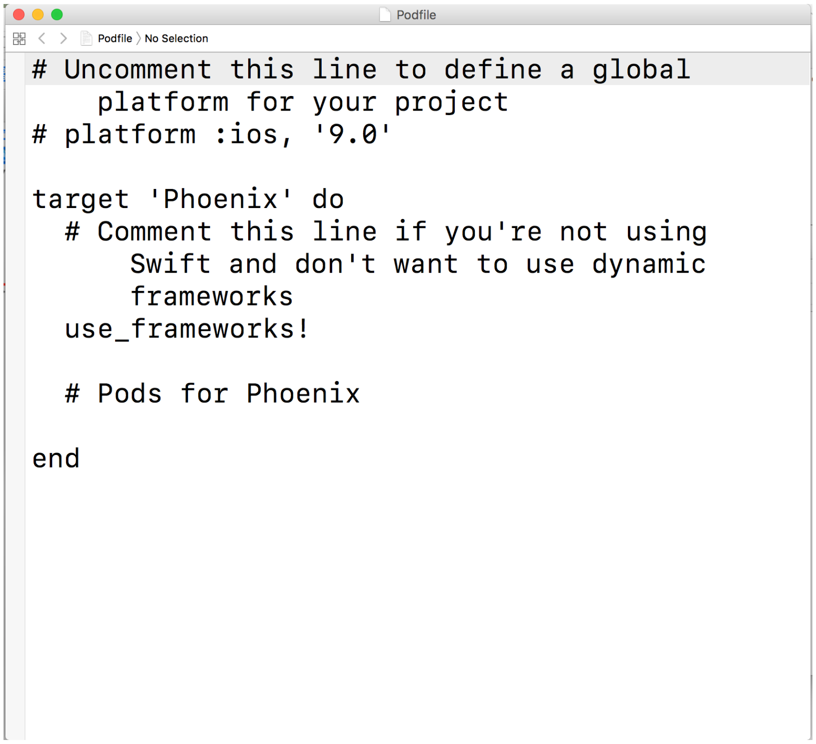
14. Once you've navigated successfully into your project, type in:

    ```
    pod init
    ```

    and press "Enter". If you check your Phoenix folder in your Finder, you'll see that Cocoapods has added something called a "Podfile" in it, like so:



15. Drag and drop the Podfile onto the XCode icon onto your Dock that will most likely be on the bottom of your screen. This should pop up:

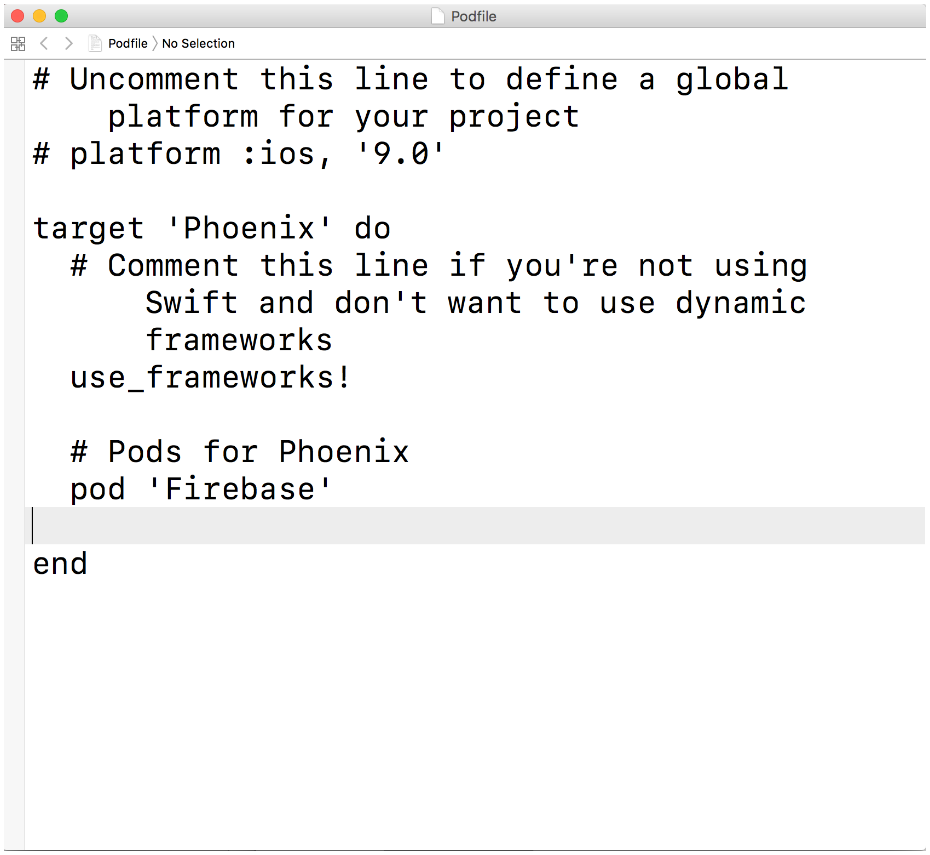


```
# Uncomment this line to define a global
    platform for your project
# platform :ios, '9.0'

target 'Phoenix' do
  # Comment this line if you're not using
      Swift and don't want to use dynamic
      frameworks
  use_frameworks!

  # Pods for Phoenix

end
```

16. Underneath the line “# Pods for Phoenix”, type in

```
pod ‘Firebase’
```

Your Podfile should look like this:



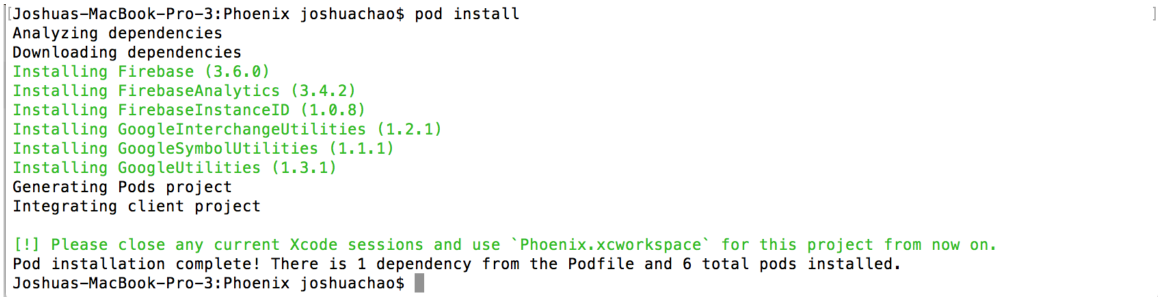
```
# Uncomment this line to define a global
    platform for your project
# platform :ios, '9.0'

target 'Phoenix' do
  # Comment this line if you're not using
      Swift and don't want to use dynamic
      frameworks
  use_frameworks!

  # Pods for Phoenix
  pod 'Firebase'

end
```

Please note the single quotes. Please save the Podfile when you are done with this change by pressing Command + S.

17. Back in Terminal, type in the following:

```
pod install
```

This will most likely take a long, long time! Really, sit back and relax, it'll take a long time to load. When it's done, it'll report the following:



```
Joshuas-MacBook-Pro-3:Phoenix joshuachao$ pod install
Analyzing dependencies
Downloading dependencies
Installing Firebase (3.6.0)
Installing FirebaseAnalytics (3.4.2)
Installing FirebaseInstanceID (1.0.8)
Installing GoogleInterchangeUtilities (1.2.1)
Installing GoogleSymbolUtilities (1.1.1)
Installing GoogleUtilities (1.3.1)
Generating Pods project
Integrating client project

[!] Please close any current Xcode sessions and use `Phoenix.xcworkspace` for this project from now on.
Pod installation complete! There is 1 dependency from the Podfile and 6 total pods installed.
Joshuas-MacBook-Pro-3:Phoenix joshuachao$
```

And that's it. You're done!!!! Hopefully that wasn't too bad. If you have any questions, just come over to the Help Desk in the middle of the venue and ask for Josh Chao (he's wearing a Nintendo cap) or Rishi Dhanaraj. We'll be happy to help you out!