

Search...

Search

- [Explore](#)
- [Gist](#)
- [Blog](#)
- [Help](#)

 [liyanchang](#)

-
- [∞](#)
-
-

- [Admin](#)
- [Pull Request](#)
- [Watch2](#) [Unwatch2](#)
- [Fork1](#)

private [Filepicker](#) / [ios](#)

- [Code](#)
- [Network](#)
- [Pull Requests 0](#)
- [Issues 0](#)
- [Wiki](#)
- [Graphs](#)

Objective-C

1. [Objective-C 100%](#)

The easiest way to import content into your application. [www.filepicker.io](#) — [Read more](#)

[Edit](#)

- [Clone in Mac](#)
- [ZIP](#)
- [HTTP](#)
- [SSH](#)

https://github.com/Fi

Read+Write access

- [Tags 0](#)
 - [Downloads 0](#)
 - Search source code...
 - [branch: master](#)

Switch branches/tags

Filter branches/tags

 - [Branches](#)
 - [Tags](#)
- Nothing to show


[master](#)

- [Files](#)
- [Commits](#)
- [Branches 1](#)

Latest commit to the **master** branch

[docs](#)

[commit 0b7850f490](#)

 [liyanchang](#) authored a minute ago
[ios](#) /

name	age	history message
Documenation Files	3 days ago	docs [liyanchang]
FilepickerDemo	3 minutes ago	docs [liyanchang]
FilepickerDemoText	3 minutes ago	docs [liyanchang]
README - Additional Documentation	10 hours ago	more things [liyanchang]
library	34 minutes ago	last changes before freedom [liyanchang]
.gitignore	a day ago	adding plist to git ignore so I can put in my apikey [liyanchang]
README.md	a minute ago	docs [liyanchang]
license.txt	3 days ago	adding library [liyanchang]
README.md		

Filepicker iOS Library

The easiest way to import content into your application. [Filepicker.io](#)

Dependancies

- Software
 - Built targeting iOS 4.3
 - XCode 4
- Frameworks
 - AssetsLibrary.framework
 - QuartzCore.framework
 - CoreFoundation.framework
 - MobileCoreServices.framework
 - CoreGraphics.framework
 - Foundation.framework
 - UIKit.framework

Usage Instructions

```
/*
 * To import the library
 */
#import <FPPicker/FPPicker.h>

/*
 * Opening Files
 */
// To create the object
FPPickerController *fpController = [[FPPickerController alloc] init];

// Set the delegate
fpController.fpdelegate = self;

// Ask for specific data types. (Optional) Default is all files.
fpController.dataTypes = [NSArray arrayWithObjects:@"text/plain", nil];
```

```
// Select and order the sources (Optional) Default is all sources
fpController.sourceNames = [[NSArray alloc] initWithObjects: FPSourceImagesearch, FPSourceDropbox, nil];

// Display it.
[self presentViewController:fpController animated:YES];

/*
 * Saving Files
 */
// To create the object
FPSaveController *fpSave = [[FPSaveController alloc] init];

// Set the delegate
fpSave.fpdelegate = self;

// Ask for specific data mimetypes. (Optional) Default is all files.
fpController.dataTypes = [NSArray arrayWithObjects:@"text/plain", nil];

// Select and order the sources (Optional) Default is all sources
//fpSave.sourceNames = [NSArray arrayWithObjects: FPSourceCamera, FPSourceCameraRoll, FPSourceDropbox, FPSourceFacebook, FPSourceGmail, FPSourceBox, FPSourceGithub, FPSourceGoogle, nil];

// Set the data and data type to be saved.
fpSave.data = [[NSData *alloc] init] ;
fpSave.dataType = @"text/plain";
//alternative: fpSave.dataExtension = @"txt"

//optional: propose the default file name
fpSave.proposedFilename = @"AwesomeFile";

// Display it.
[self presentViewController:fpController animated:YES];
```

Delegate Functions

FPPickerControllerDelegate

```
- (void)FPPickerController:(FPPickerController *)picker didFinishPickingMediaWithInfo:(NSDictionary *)info;
```

- Keys in the info dictionary
 - FPPickerControllerMediaType
 - the UTType of the file
 - e.g: public.image
 - FPPickerControllerReferenceURL
 - The local location of the file.
 - e.g: assets-library://asset/asset.JPG?id=1000000001&ext=JPG
 - FPPickerControllerRemoteURL
 - The URL for the file.
 - e.g: <https://www.filepicker.io/api/file/we9f3kf93qls0>
 - These will be active for 4 hours or permanent, depending on your filepicker plan
 - (When Possible) FPPickerControllerOriginalImage
 - The UIImage
 - e.g: <UIImage: 0x8a37730>

```
- (void)FPPickerControllerDidCancel:(FPPickerController *)picker
```

FPSaveControllerDelegate Methods

```
- (void)FPSaveController:(FPSaveController *)picker didFinishPickingMediaWithInfo:(NSDictionary *)info;
```

- IMPORTANT NOTE: Info is an empty dictionary; nothing is being passed back

```
- (void)FPSaveControllerDidCancel:(FPSaveController *)picker;
```

List of all sources:

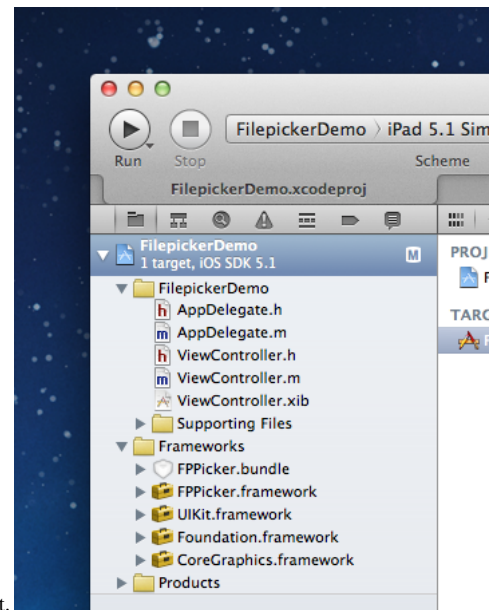
- Complete Listing

- FPSourceCamera
 - The Local Camera
 - Open: "image/jpeg", "image/png"
 - Save: nil
- FPSourceCameraRoll
 - The Local Photos
 - Open: "image/jpeg", "image/png", "video/quicktime"
 - Save: "image/jpeg", "image/png"
- FPSourceDropbox
 - www.dropbox.com
 - Open: "*"/*"
 - Save: "*"/*"
- FPSourceFacebook
 - www.facebook.com
 - Open: "image/jpeg"
 - Save: "image/*"
- FPSourceGmail
 - www.gmail.com
 - Open: "*"/*"
 - Save: nil
- FPSourceBox
 - www.box.com
 - Open: "*"/*"
 - Save: "*"/*"
- FPSourceGithub
 - www.github.com
 - Open: "*"/*"
 - Save: nil
- FPSourceGoogleDrive
 - drive.google.com
 - Open: "*"/*"
 - Save: "*"/*"
- FPSourceImagesearch
 - Flickr Public Domain Image Search
 - Open: "image/jpeg"
 - Save: nil

Installation Instructions

For iOS Pros:

1. Get an API Key
 - Go to Filepicker.io to register an account
 - Api Keys are typically randomized and 20 characters long.
2. Insert the framework and bundle
 - Download or clone the repository.
 - Under /library, you'll find `FPPicker.framework` and `FPPicker.bundle`
 - Drag both into your project, typically in your framework folder



- `#import <FPPicker/FPPicker.h>` in your `viewController.h` or other file where you want to use it.

3. Settings

- In your application's info.plist, add the following key/value:

Key: "Filepicker API Key"

Value: YOUR_API_KEY (that you got from step 1)

- If this doesn't build, I've had luck adding `-all_load -ObjC` in `Build_Settings/Other_Linker_Flags`
- You may need to add additional frameworks
 - `AssetsLibrary.framework`
 - `QuartzCore.framework`
 - `CoreFoundation.framework`
 - `MobileCoreServices.framework`
 - `CoreGraphics.framework`
 - `Foundation.framework`
 - `UIKit.framework`

4. Use it

- Initialize it:

```
FPPickerController *fpController = [[FPPickerController alloc] init];
fpController.fpdelegate = self;
- Delegate Methods:
- (void)FPPickerController:(FPPickerController *)picker didFinishPickingMediaWithInfo:(NSDictionary *)info;
- (void)FPPickerControllerDidCancel:(FPPickerController *)picker;
```

Running the Demo Project:

1. Get an Api Key:

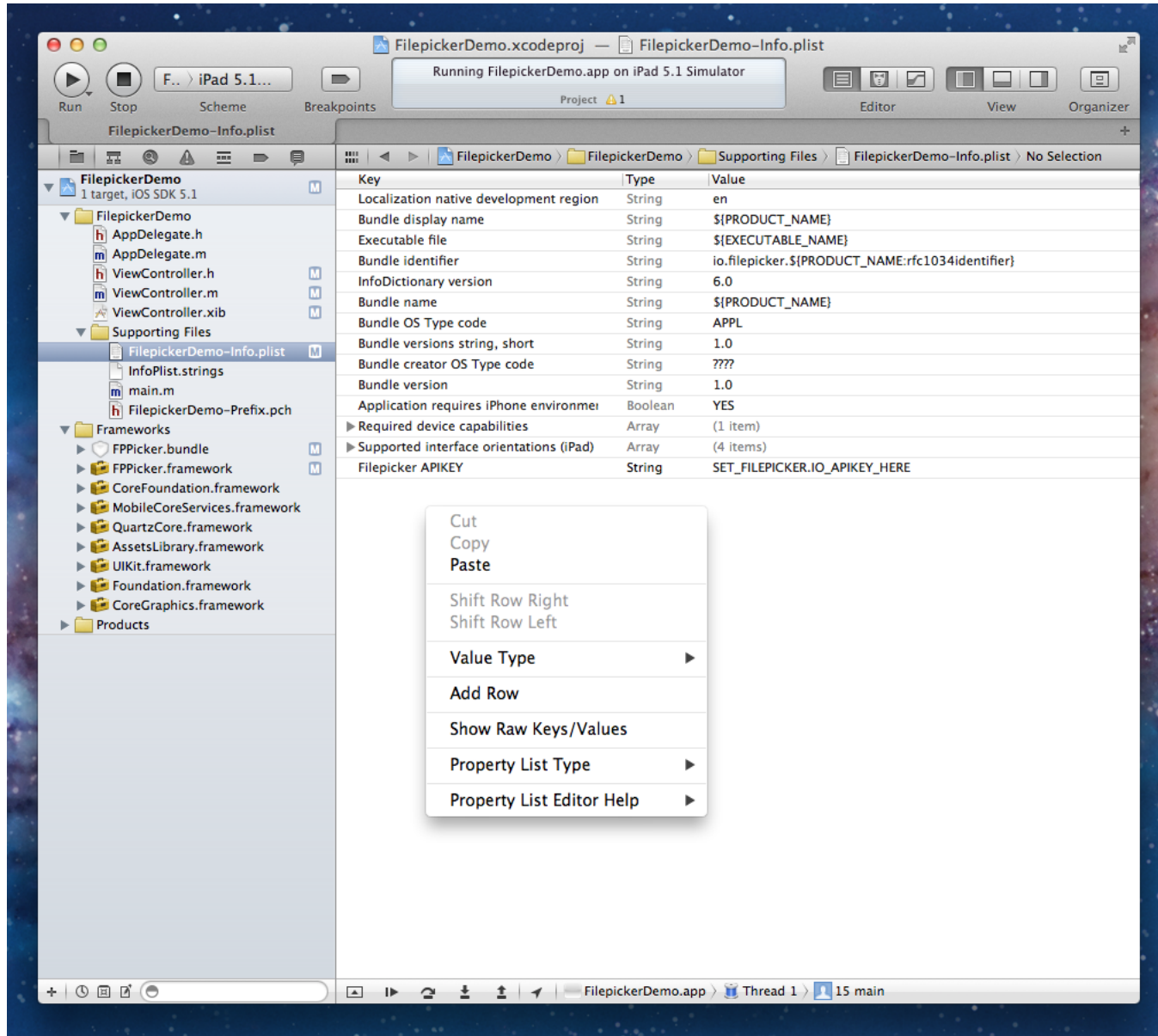
- Go to Filepicker.io to register an account
- Api Keys are typically randomized and 20 characters long.

2. Insert the framework and bundle:

- Download or clone the repository.
- Open the `/Examples/FPDemo.xcodeproj`

3. Settings:

- In your application's info.plist, add the following key/value:



Key: "Filepicker API Key"
Value: YOUR_API_KEY (that you got from step 1)

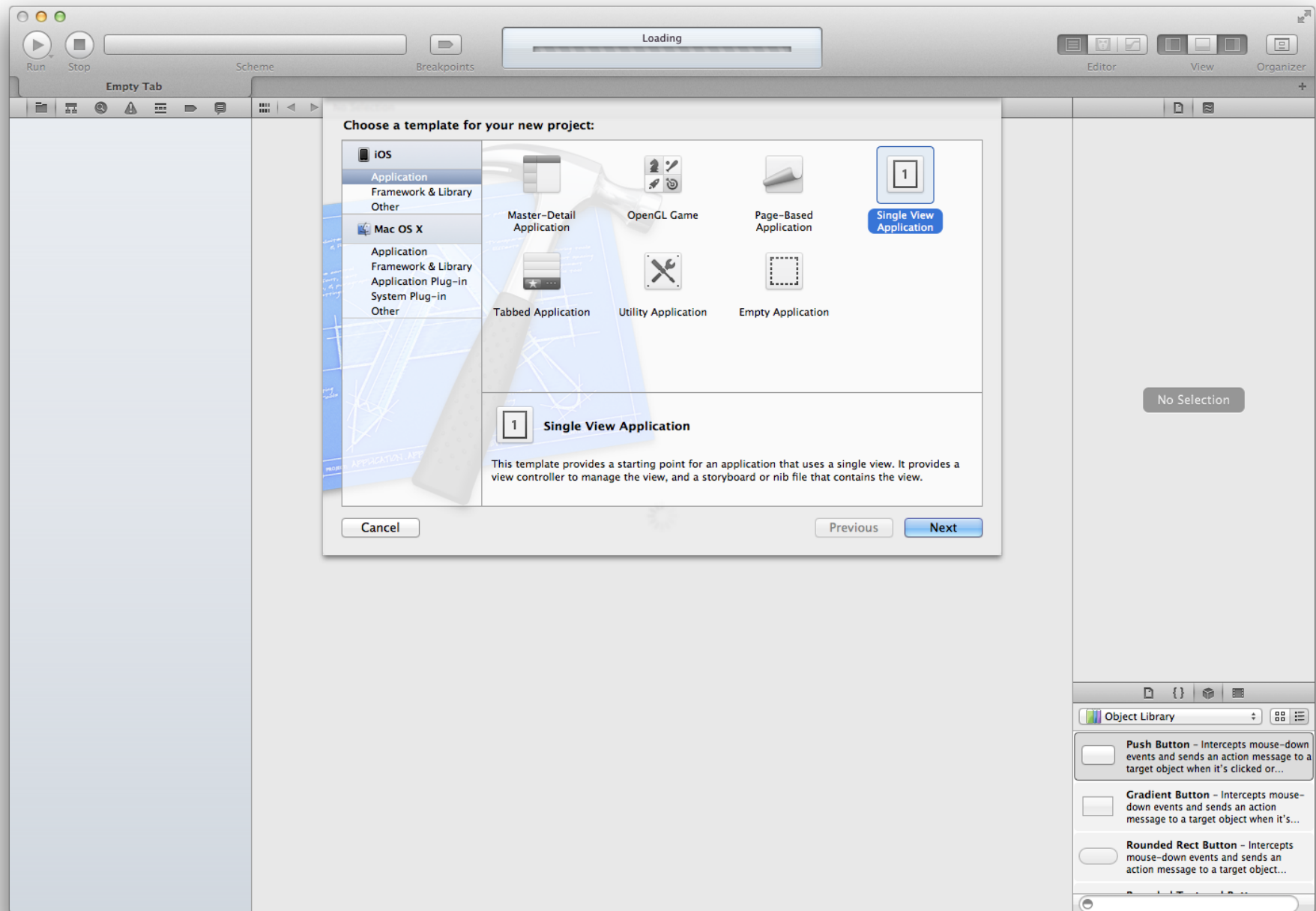
4. Build and Run

Starting from scratch:

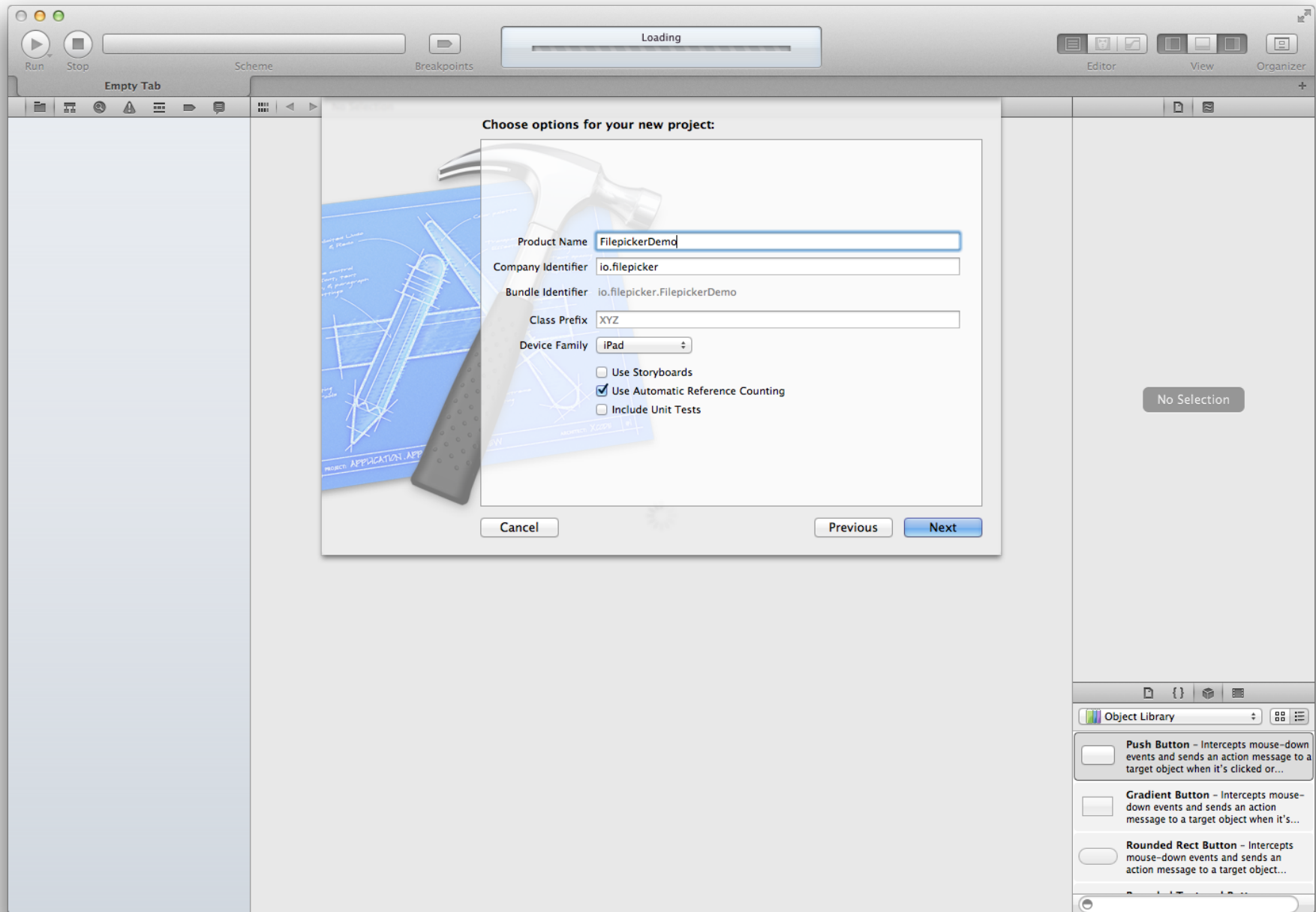
1. Get an Api Key:

- Go to [Filepicker.io](https://filepicker.io) to register an account
- Api Keys are typically randomized and 20 characters long.

2. Start a new Project

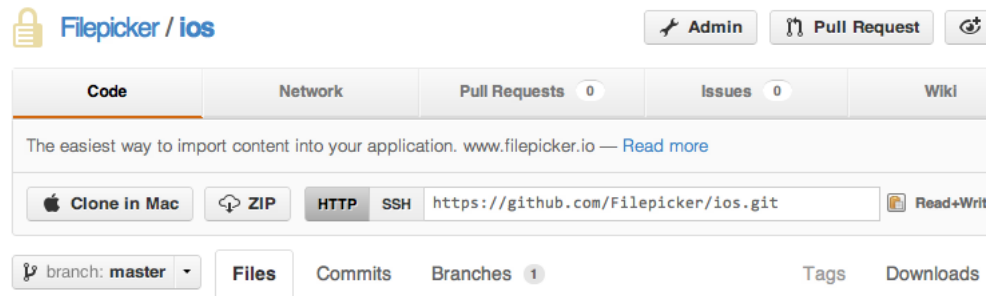


- File/New/Project or Shift-Apple N
- Single View Project

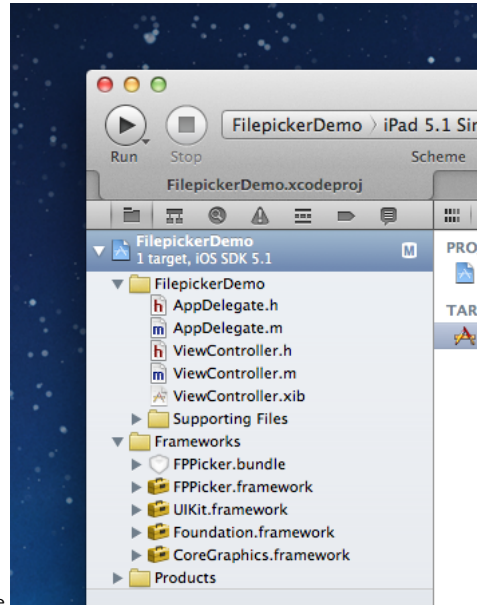


- Name: FilepickerDemo
- Device Family: iPad
- Use Storyboards: False
- Use Automatic Reference Counting: True

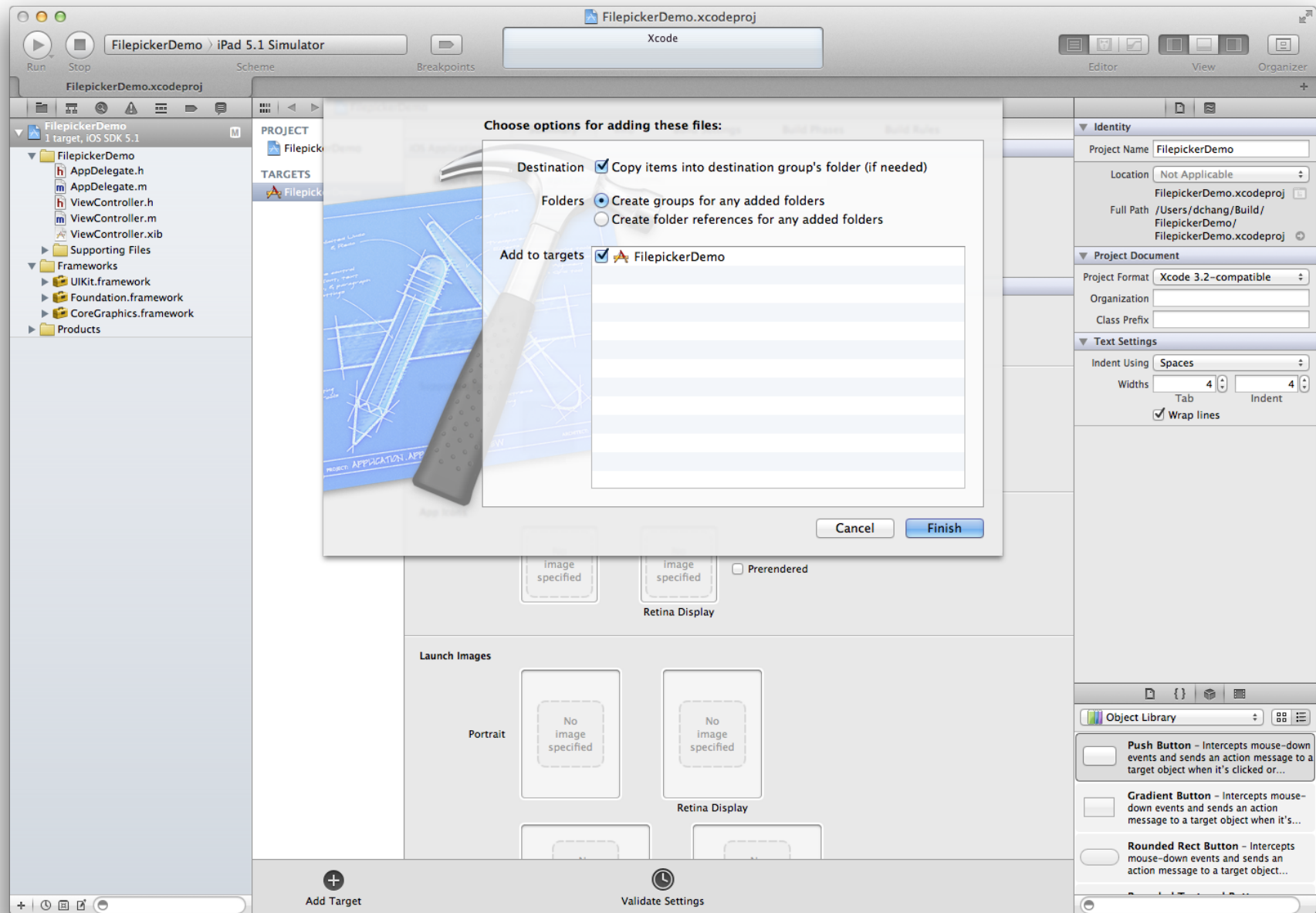
3. Insert the framework and bundle



- Download or clone the repository.
- You can do this at the top of this git repository with either zip to get the zip or git clone <https://github.com/Filepicker/ios.git>.
- Open up the folder.

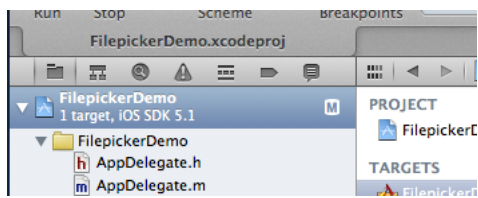


- Under library/, you'll find FPPicker.framework and FPPicker.bundle
- Drag both into your project, typically in your framework folder

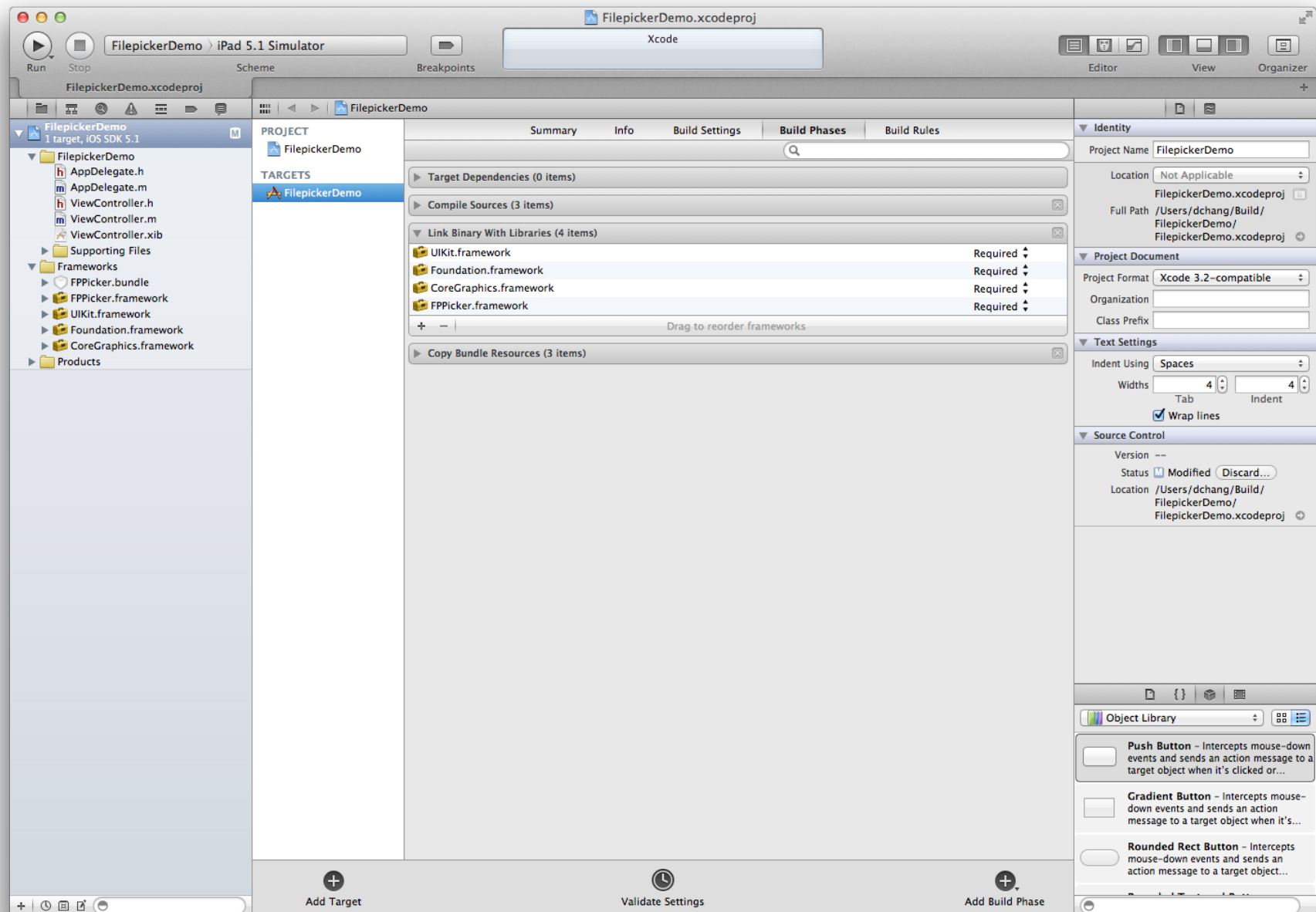


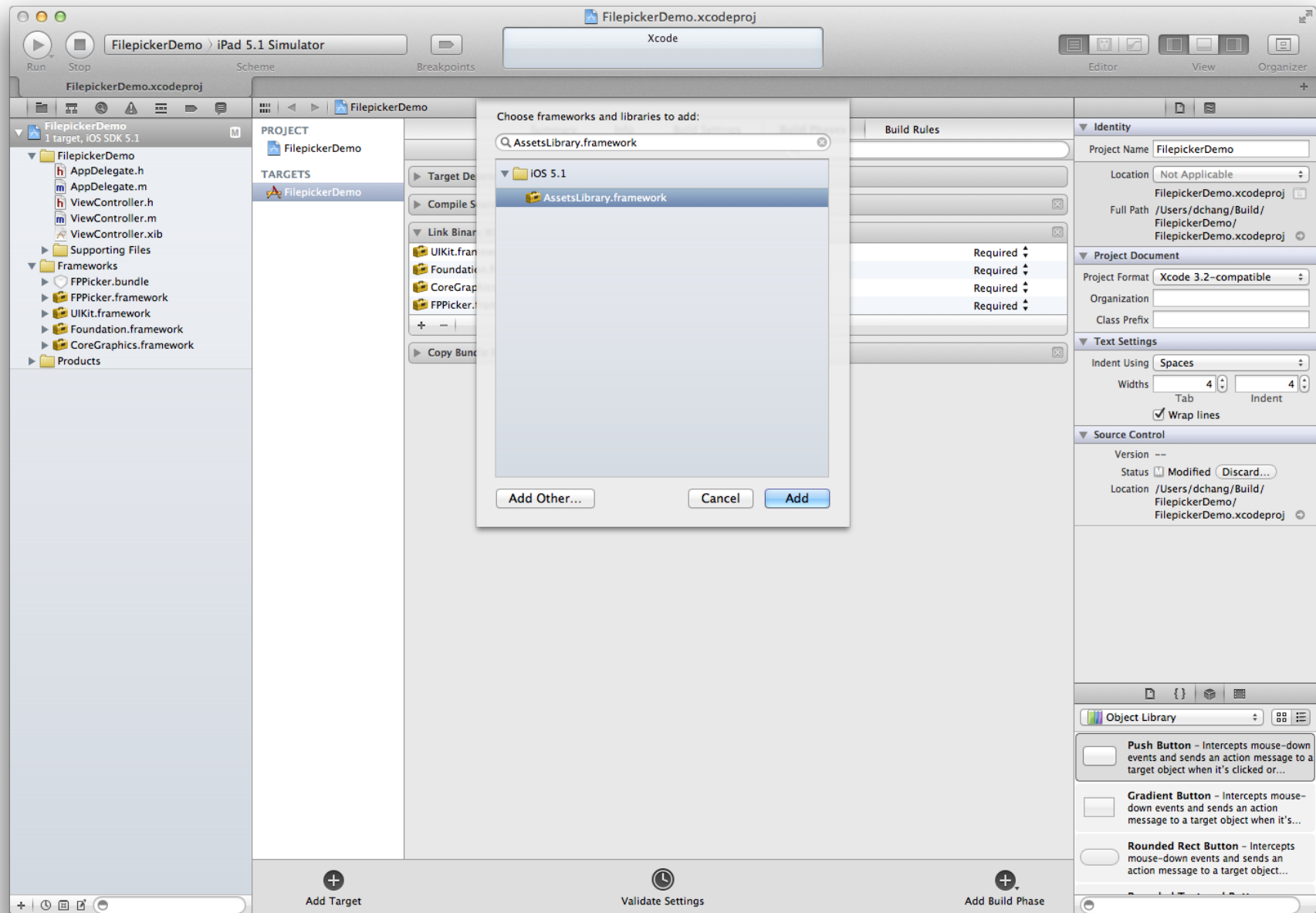
- Choose to Copy items into Destination Folder

4. Adding Additional Frameworks



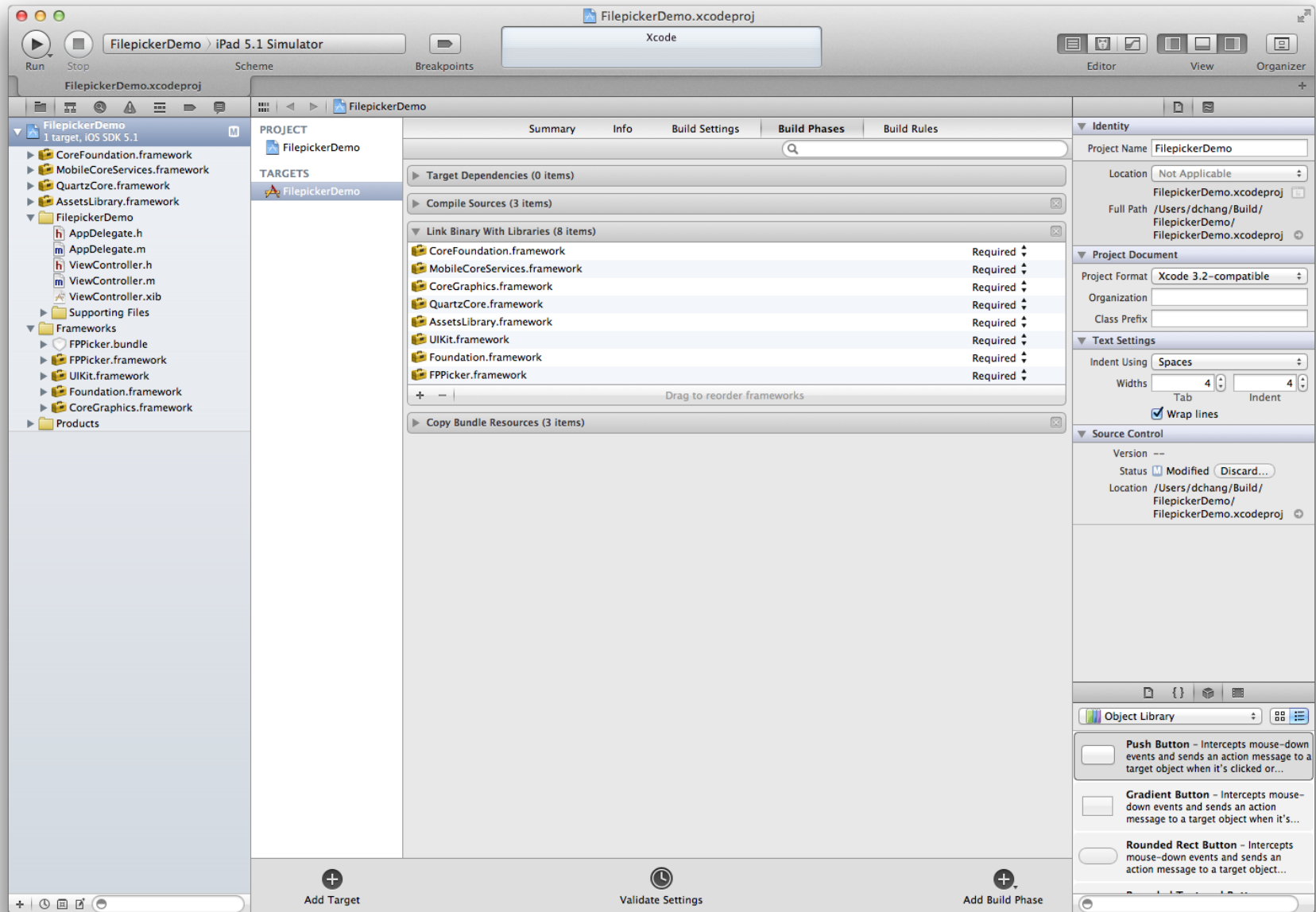
- Click on the .xcodeProj



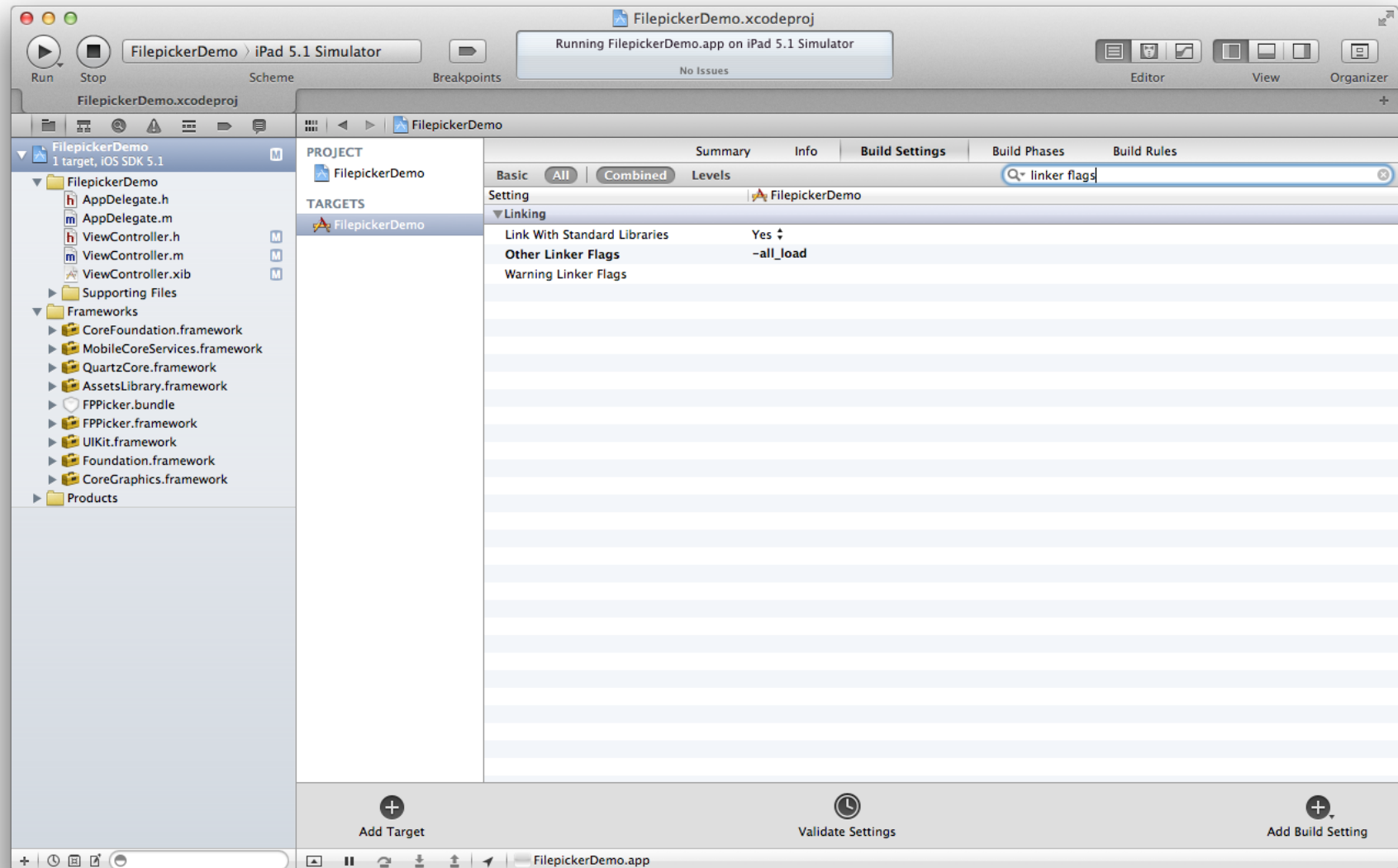


o Under Build Phases -> Link Binary with Libraries, add the following:

- AssetsLibrary.framework
- QuartzCore.framework
- CoreGraphics.framework
- MobileCoreServices.framework
- Foundation.framework
- CoreFoundation.framework



- Under Build Settings, search for Other Linker Flags and set it to `-all_load`

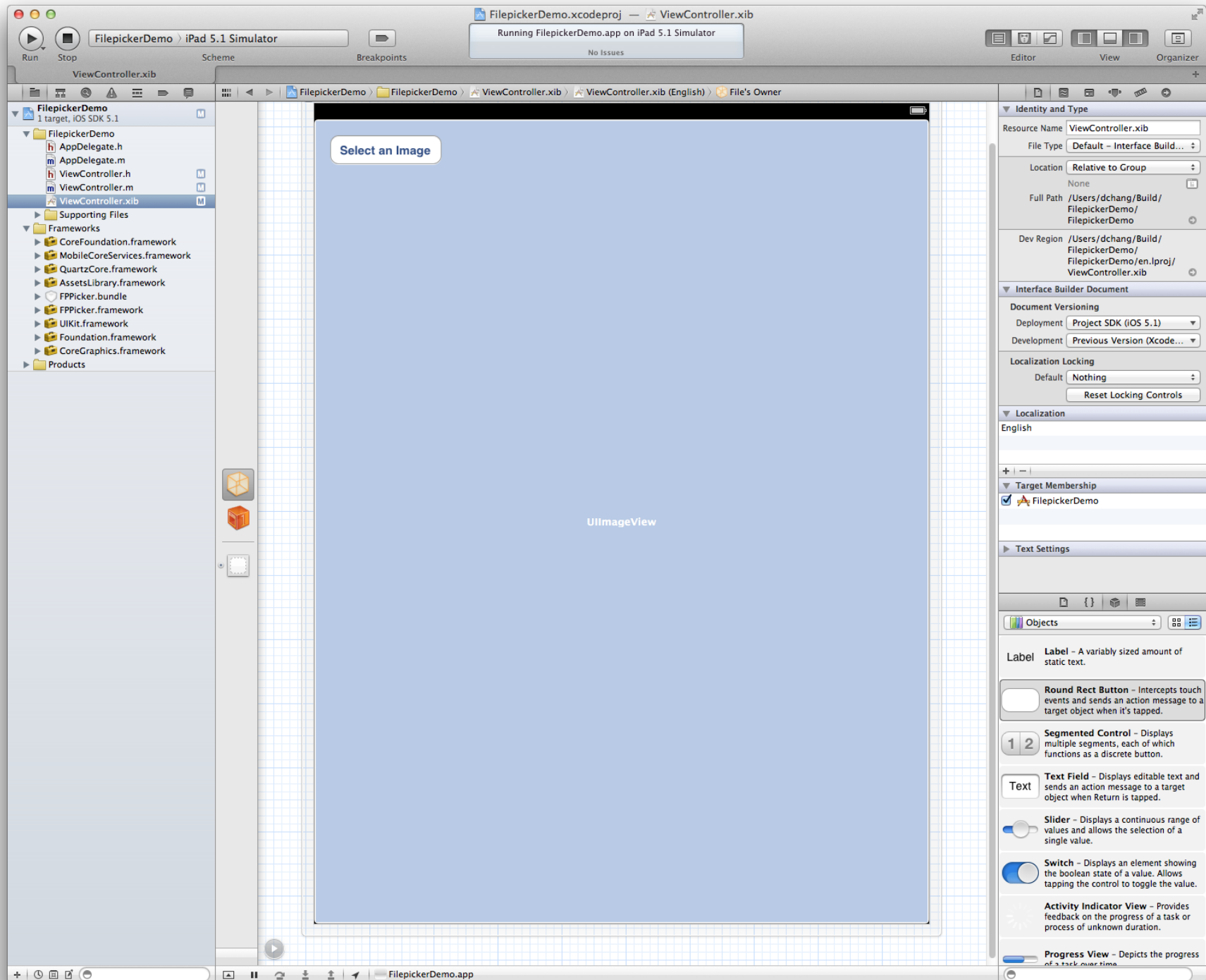


5. Write the code

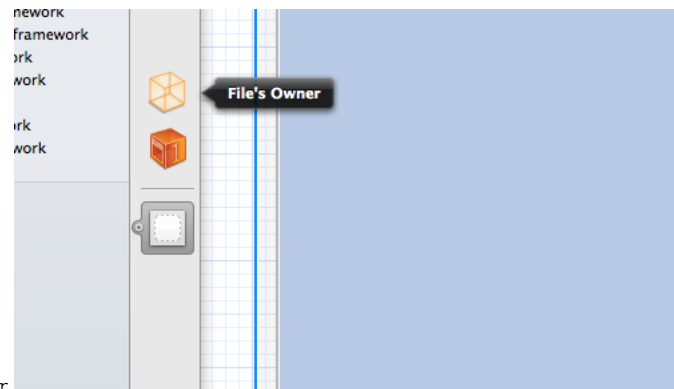
- You can copy and paste the following code into your `viewController.h` and `viewController.m` respectively.
- In `viewController.h`
 - We create a button, imageview, and a popover
- In `viewController.m`
 - We create an action when the button is pressed
 - We also have two delegates that respond when the Filepicker is finished.
- [ViewController.h Source](#)
- [ViewController.m Source](#)

6. Build the User Interface

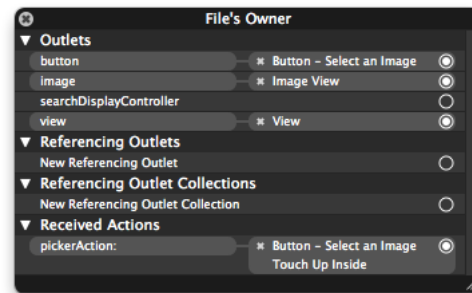
- Drag a 'Image View' and a 'Rounded Rectangle Button' from the objects draw on the right.



- Hook up the proper interfaces



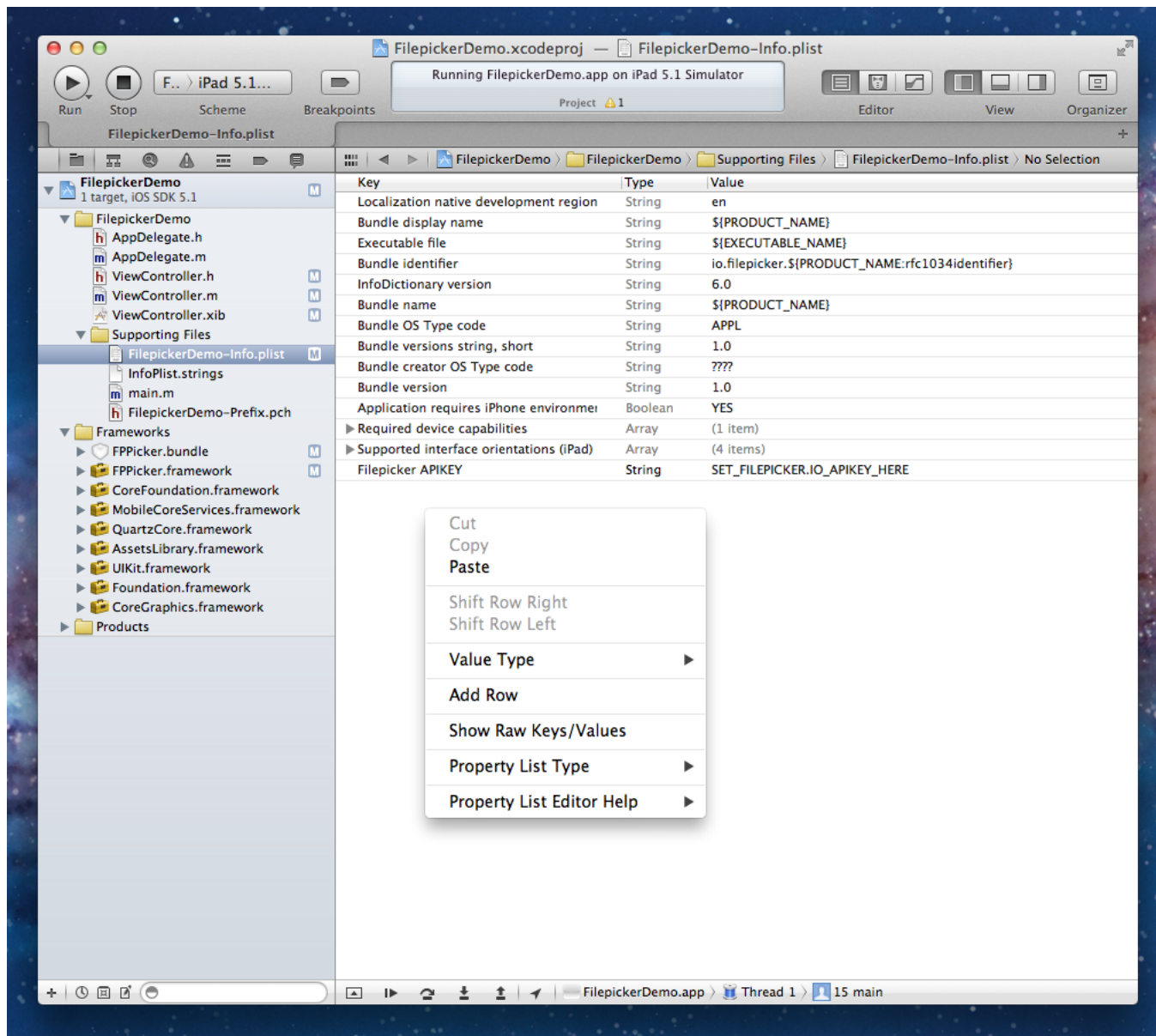
- On the right hand side, you'll notice three icons, one of which is the File's Owner
- Hold down Ctrl, click and hold down the File Owner Orange Cube, then drag to the Image. A small popup will ask you to connect the file owner to the image. Choose outlet: Image.
- In a similar manner, ctrl drag from File Owner -> Button. Choose outlet: Button.
- Now, in the opposite direction, ctrl dragging from the button to the file owner. Choose pickerAction:.



- Now if you right click on file owner, you should see the following.

7. Add your API Key

- Go to Supporting Files/FilepickerDemo-Info.plist. (Your's may vary if you didn't name it FilepickerDemo).
- Right click, Add Row.
- For the key: Filepicker API Key
- For the value, paste in your apikey that you got from filepicker.io



8. Run.

- Click the run button in the upper right corner of xcode.



- It should build and you can now choose a file!

GitHub Links

GitHub

- [About](#)
- [Blog](#)
- [Features](#)
- [Contact & Support](#)
- [Training](#)
- [GitHub Enterprise](#)
- [Site Status](#)

Clients

- [GitHub for Mac](#)
- [GitHub for Windows](#)
- [GitHub for Eclipse](#)
- [GitHub Mobile Apps](#)

Tools

- [Gauges: Web analytics](#)
- [Speaker Deck: Presentations](#)
- [Gist: Code snippets](#)

Extras

- [Job Board](#)
- [GitHub Shop](#)
- [The Octodex](#)

Documentation

- [GitHub Help](#)
- [Developer API](#)
- [GitHub Flavored Markdown](#)
- [GitHub Pages](#)
- [Terms of Service](#)
- [Privacy](#)
- [Security](#)

© 2012 GitHub Inc. All rights reserved.



Powered by the [Dedicated Servers](#) and [Cloud Computing](#) of Rackspace Hosting®

Markdown Cheat Sheet

Format Text

Headers

```
# This is an <h1> tag
## This is an <h2> tag
##### This is an <h6> tag
```

Text styles

This text will be italic
This will also be italic
****This text will be bold****
__This will also be bold__

*You ****can**** combine them*

Lists

Unordered

- * Item 1
- * Item 2
 - * Item 2a
 - * Item 2b

Ordered

1. Item 1
2. Item 2
3. Item 3
 - * Item 3a
 - * Item 3b

Miscellaneous

Images

![[GitHub Logo]](/images/logo.png)
Format: ![Alt Text](url)

Links

<http://github.com> - automatic!
[GitHub](http://github.com)

Blockquotes

As Kanye West said:

> We're living the future so
> the present is our past.

Code Examples in Markdown

Syntax highlighting with [GFM](#)

```
```javascript
function fancyAlert(arg) {
 if(arg) {
 $.facebox({div:'#foo'})
 }
}
```
```

Or, indent your code 4 spaces

Here is a Python code example
without syntax highlighting:

```
def foo:
    if not bar:
        return true
```

Inline code for comments

I think you should use an
`<addr>` element here instead.

Something went wrong with that request. Please try again. [Dismiss](#)

Looking for the GitHub logo?

- GitHub Logo



[Download](#)

- The Octocat



[Download](#)