

iPhone App Dev

Lesson 5

Source Codes

<https://github.com/makzan/ios-dev-course-example>

Contact

makzan@42games.net

Exercise

- What app are you developing?
- Any ideas worth discussing?
- Any great app design worth sharing?

Summary

- Loading network image with AFNetworking
- Introducing Category
- Declaring Category methods
- Querying Facebook profile pic
- Accessing NSDictionary
- UIScrollView
- Arranging Elements with UIScrollView

Loading Network Image with AFNetworking

Including AFNetworking

A screenshot of a Google search results page for the query "afnetworking". The search bar at the top contains "afnetworking". The results show approximately 30,800 results found in 0.31 seconds. The first result is a link to the GitHub repository for AFNetworking, titled "AFNetworking/AFNetworking · GitHub". Below the link, it says "https://github.com/AFNetworking/AFNetworking - Cached" and provides a snippet of the description: "1 day ago – A delightful iOS and OS X networking framework. Contribute to AFNetworking development by creating an account on GitHub." To the right of the snippet is a "More" button. To the left of the main search results is a sidebar with links for "Everything", "Images", "Maps", "Videos", "News", "Shopping", "More", and "Show search tools". Other visible results include "Getting Started with ...", "Pull Requests 4", "Commits", "(AFNetworking) · GitHub", "AFNetworking FAQ", and "AFOAuth2Client". At the bottom of the results, there is a link "More results from github.com »".

Search the **AFNetworking** code in github.

Including AFNetworking

The screenshot shows the GitHub repository page for the AFNetworking framework. The URL in the address bar is <https://github.com/AFNetworking/AFNetworking>. The repository name is displayed prominently at the top left. On the right side of the header, there are links for GitHub, Inc., Reader, and a search bar containing 'afnetwork'. Below the header, there's a navigation bar with links for Signup and Pricing, Explore GitHub, Features, Blog, and Login. The main content area starts with a brief description: 'A delightful iOS and OS X networking framework — [Read more](#)' followed by a link to the documentation: <http://afnetworking.org/Documentation/>. Below this, there are download options: Clone In Mac, ZIP, HTTP, Git Read-Only, and a link to the GitHub page (<https://github.com/AFNetworking/AFNetworking.git>). There's also a 'Read-Only access' button. At the bottom of the page, there's a commit history section showing a merge pull request from shanev/doc_fix, authored by mattt 2 days ago, with a commit hash of 8657907f1a.

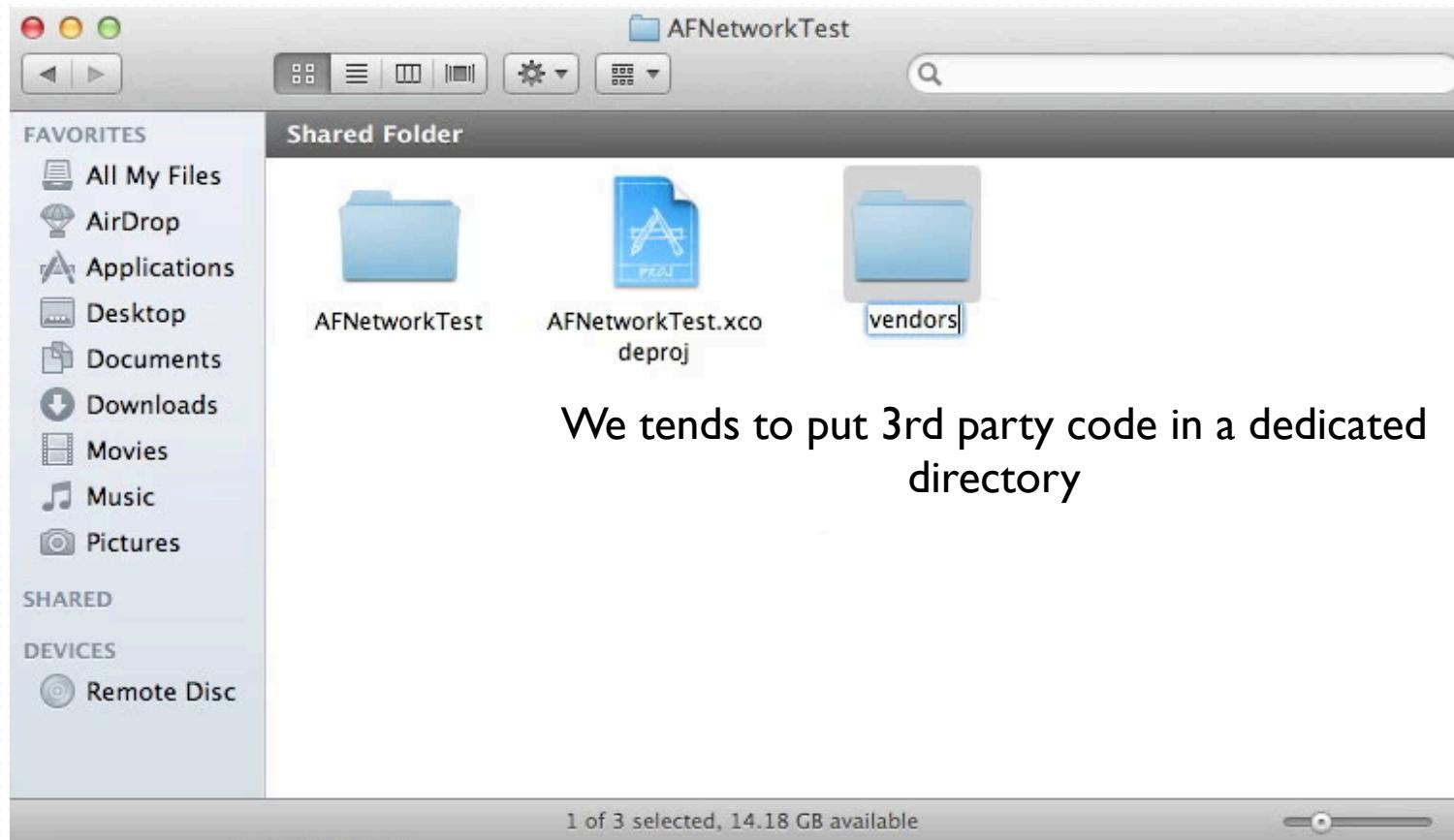
Click in the download button

Including AFNetworking

The screenshot shows the GitHub repository page for 'AFNetworking / AFNetworking'. At the top, there's a navigation bar with links for 'Signup and Pricing', 'Explore GitHub', 'Features', 'Blog', and 'Login'. Below that, the repository name 'AFNetworking / AFNetworking' is displayed, along with 'PUBLIC' status, a 'Watch' button (2,804), a 'Fork' button (378), and a star icon. A tab bar below shows 'Code' (selected), 'Network', 'Pull Requests (1)', 'Issues (6)', 'Wiki', and 'Graphs'. Underneath, there are links for 'Files', 'Commits', 'Branches (3)', 'Tags (13)', and 'Downloads'. Two download buttons are visible: 'Download as zip' (highlighted with a cursor) and 'Download as tar.gz'. A note at the bottom states: 'There aren't any downloads yet. But don't worry! You can download the source code as a zip or tarball above.'

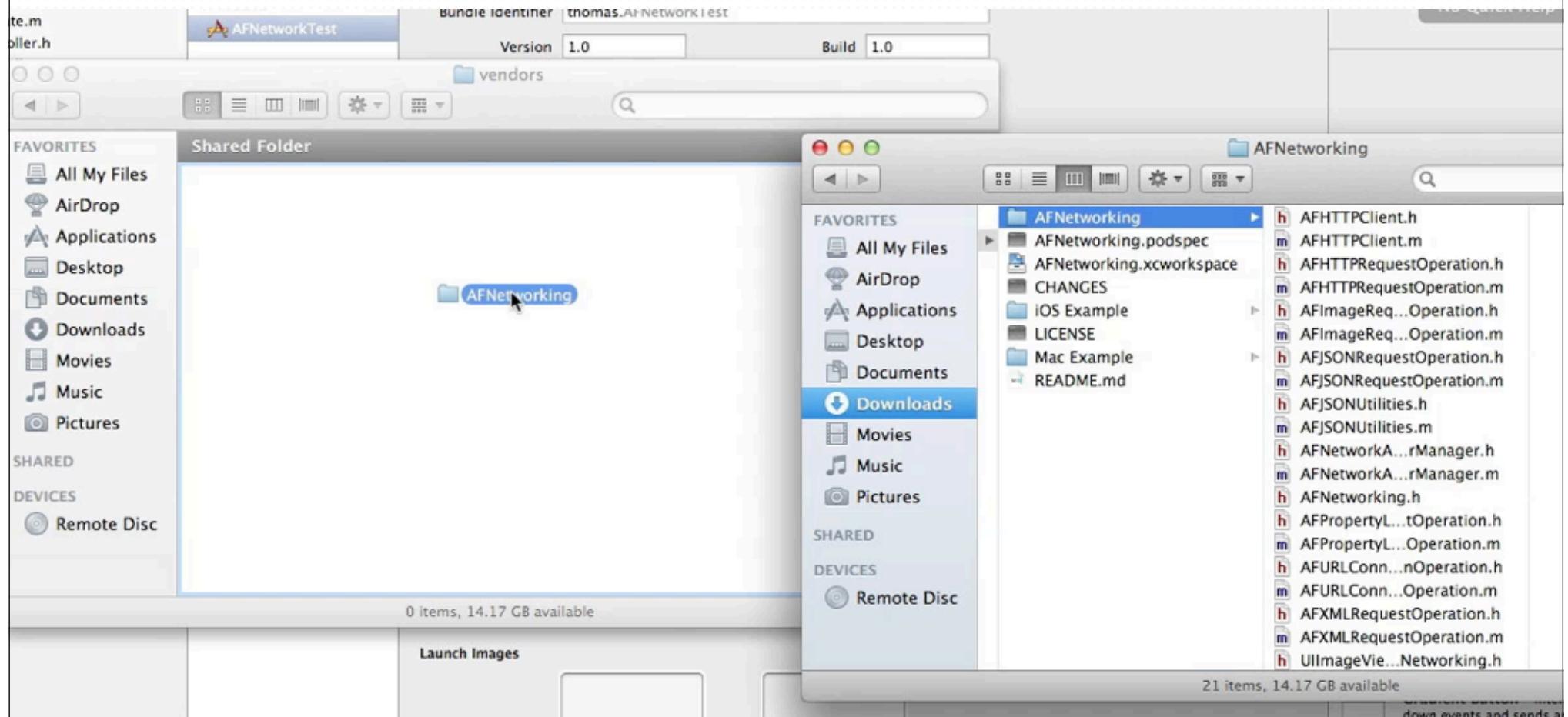
Download the code in either zip or tar

Including AFNetworking



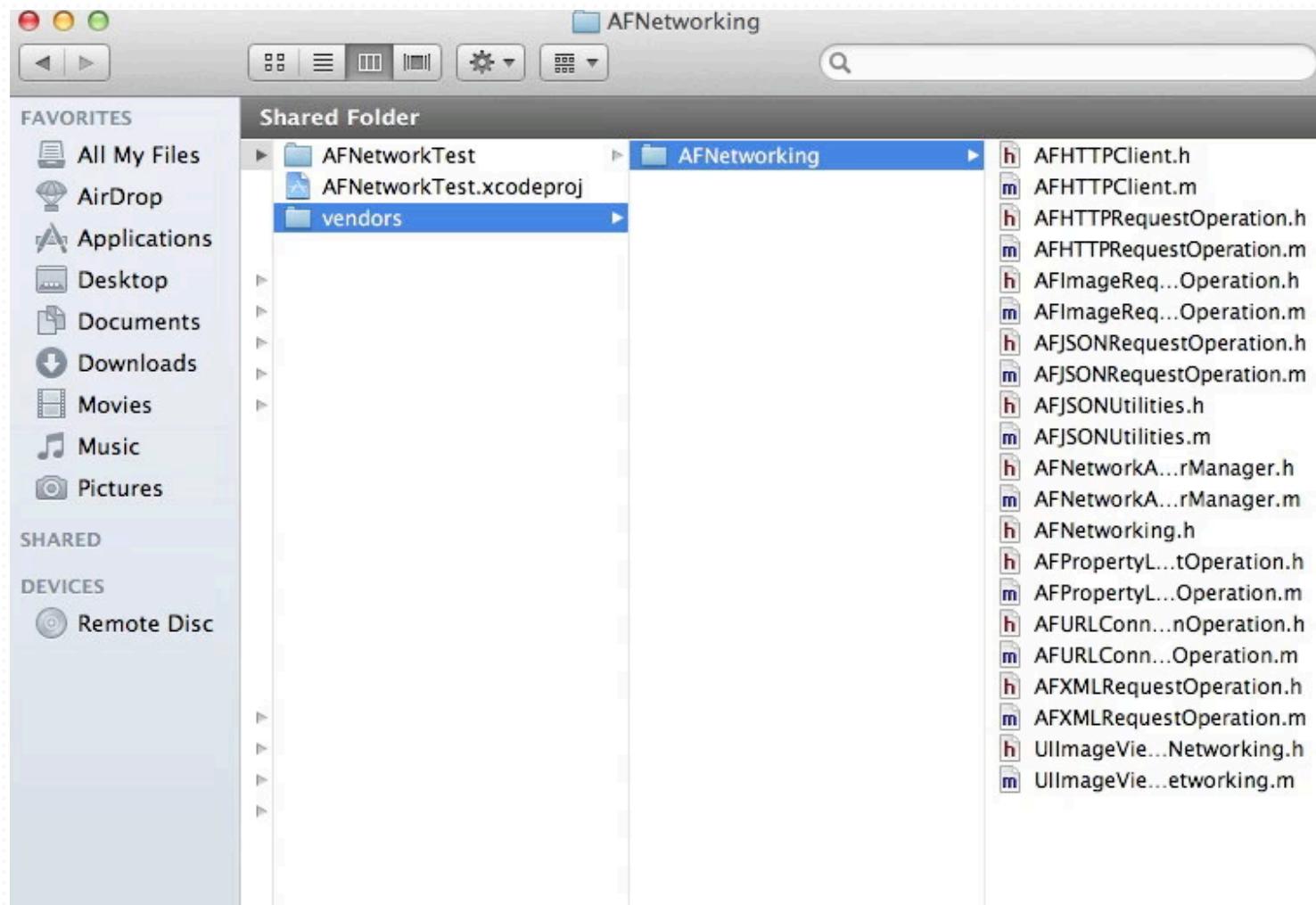
Prepare a folder named **vendors** for 3rd party code

Including AFNetworking



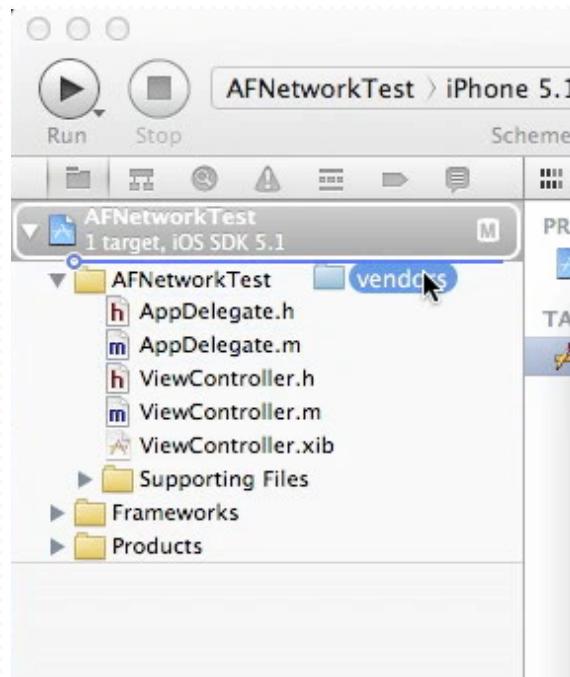
Put the AFNetworking source code into vendors

Including AFNetworking



This is what our vendors folder looks like

Including AFNetworking



Add the vendors to XCode project tree

Loading Network Images

The image shows a screenshot of the Xcode IDE. On the left, a storyboard preview window displays a single blue **Image View** component. A context menu is open over the image view, with the "Referencing Outlets" option selected. A blue circular selection dot is placed on the "New Referencing Outlet" button in the menu. On the right, the **ViewController.h** file is open in the code editor. The code includes standard header comments and imports, followed by the **@interface** section of the ViewController class. A blue line connects the "New Referencing Outlet" button in the storyboard's context menu to the **@end** keyword in the code editor, indicating that a new outlet is being automatically generated. A tooltip labeled "Insert Outlet" is visible near the connection point.

```
// ViewController.h
// AFNetworkTest
//
// Created by Freshman on 5/24/12.
// Copyright (c) 2012 __MyCompanyName__. All rights reserved.

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController
@end
```

Prepare an **UIImageView** in interface

Loading Network Images

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController
@property (retain, nonatomic) IBOutlet UIImageView *
    networkImageView;

@end
```

In this example, we named it `networkImageView`

Loading Network Images

```
#import "ViewController.h"

#import "UIImageView+AFNetworking.h"

@interface ViewController : UIViewController

@end

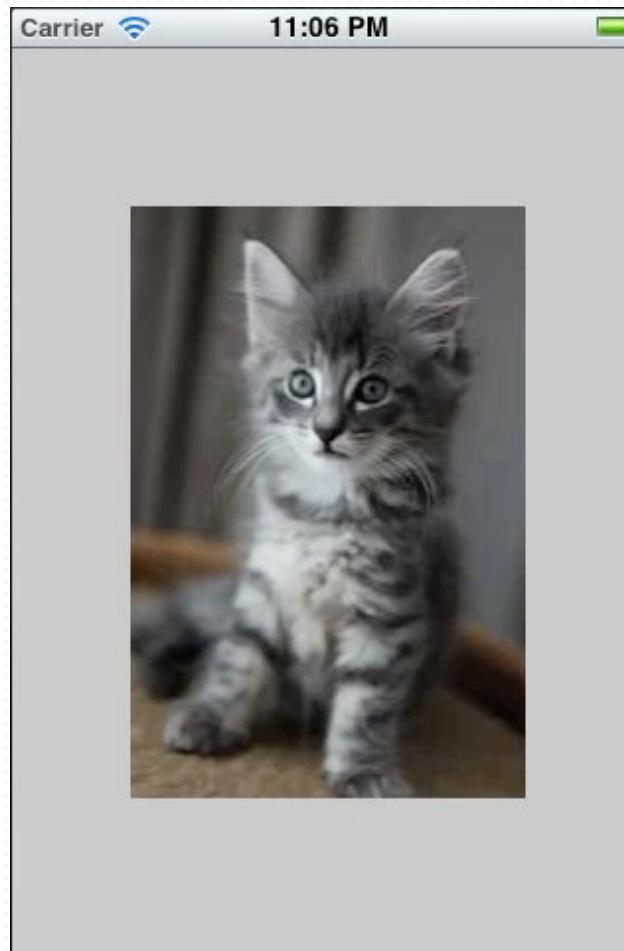
@implementation ViewController
@synthesize networkImageView;

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    NSURL * imageURL = [NSURL URLWithString:@"http://placekitten.com/200/300"];
    [networkImageView setImageWithURL:imageURL];
}
```

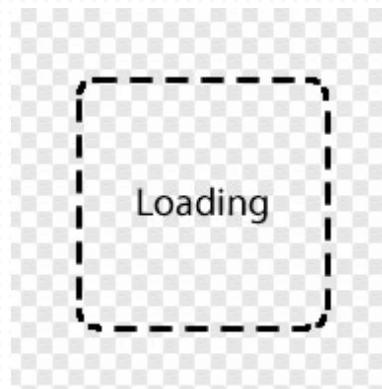
Import the UIImageView+Network.h and set the image URL

Loading Network Images



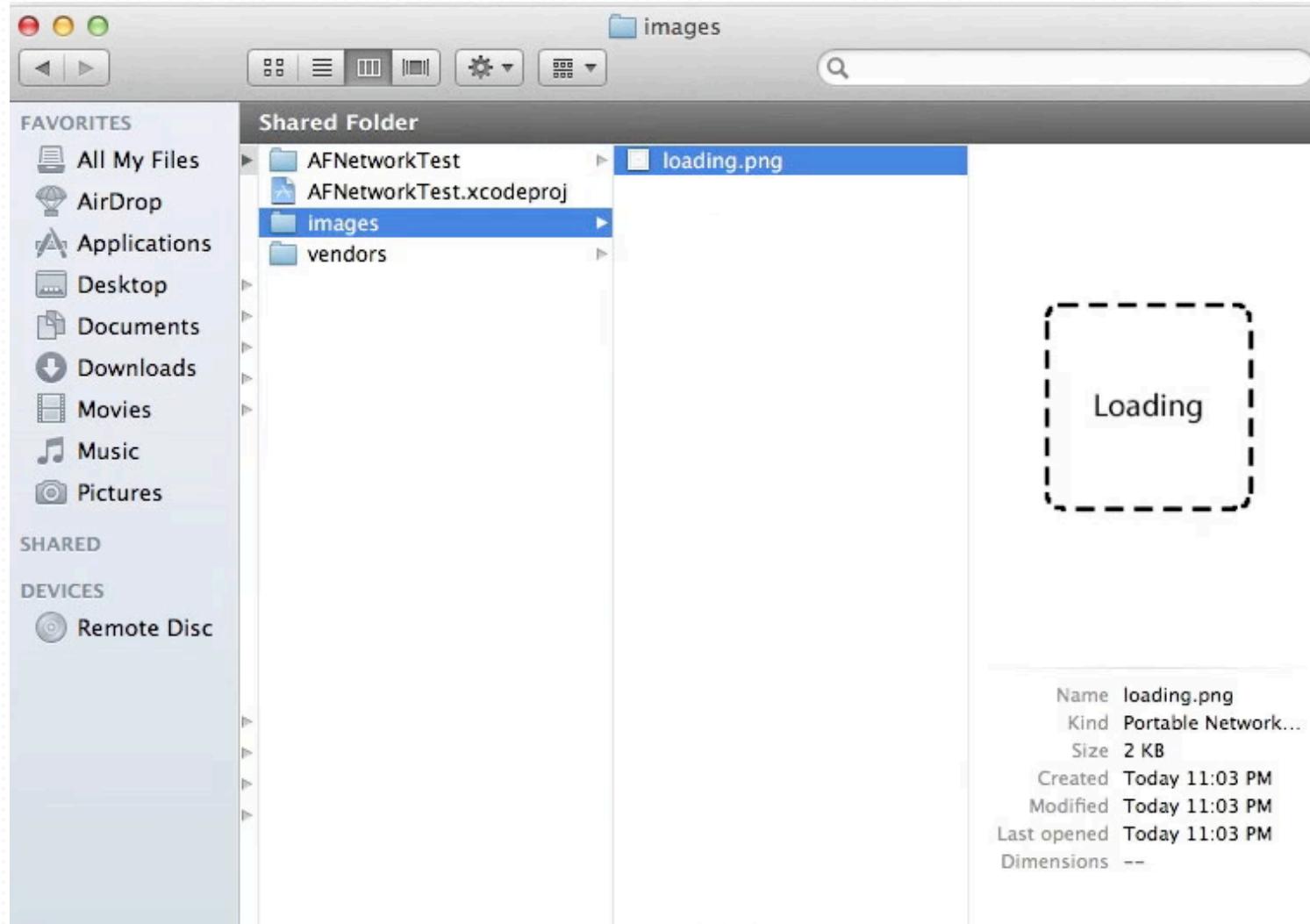
Run it in simulator and we get the image from network.

Loading Placeholder



Prepare an image to show before the target image is loaded

Loading Placeholder



Put the placeholder image into project

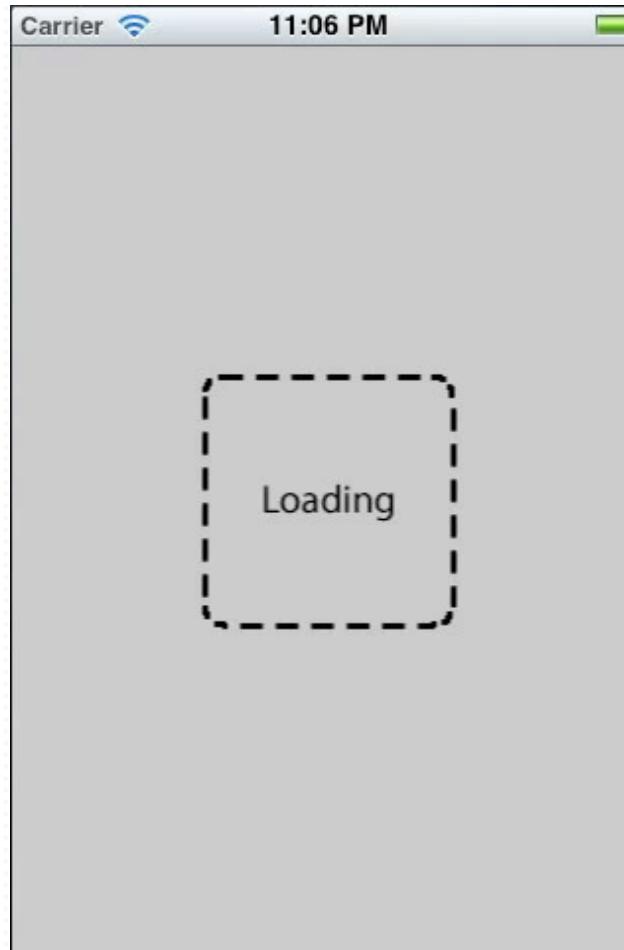
Loading Placeholder

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    NSURL * imageURL = [NSURL URLWithString:@"http://placekitten.com/200/300"];
    [networkImageView setImageWithURL:imageURL placeholderImage:[UIImage imageNamed:@"loading.png"]];
}
```

Modify the network image code with placeholder option

Loading Placeholder



The placeholder show up before the image loaded in simulator.

Introducing Category

What is Category

- It adds methods behaviors to Class at Run Time

- It is usually named in following format:

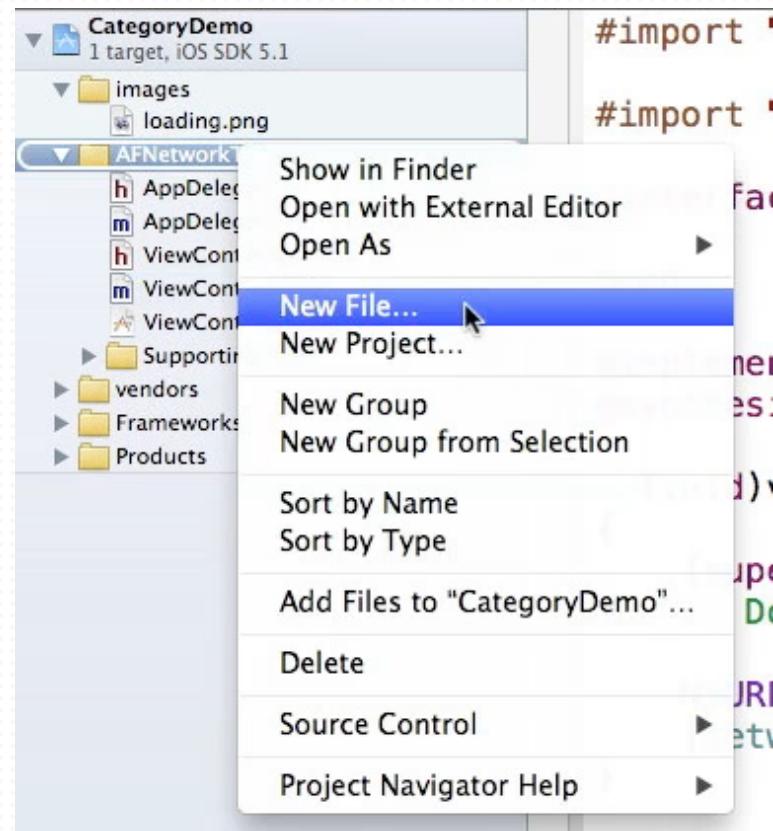
UIView+Position

- and the interface and implementation is

@interface UIView (Position)

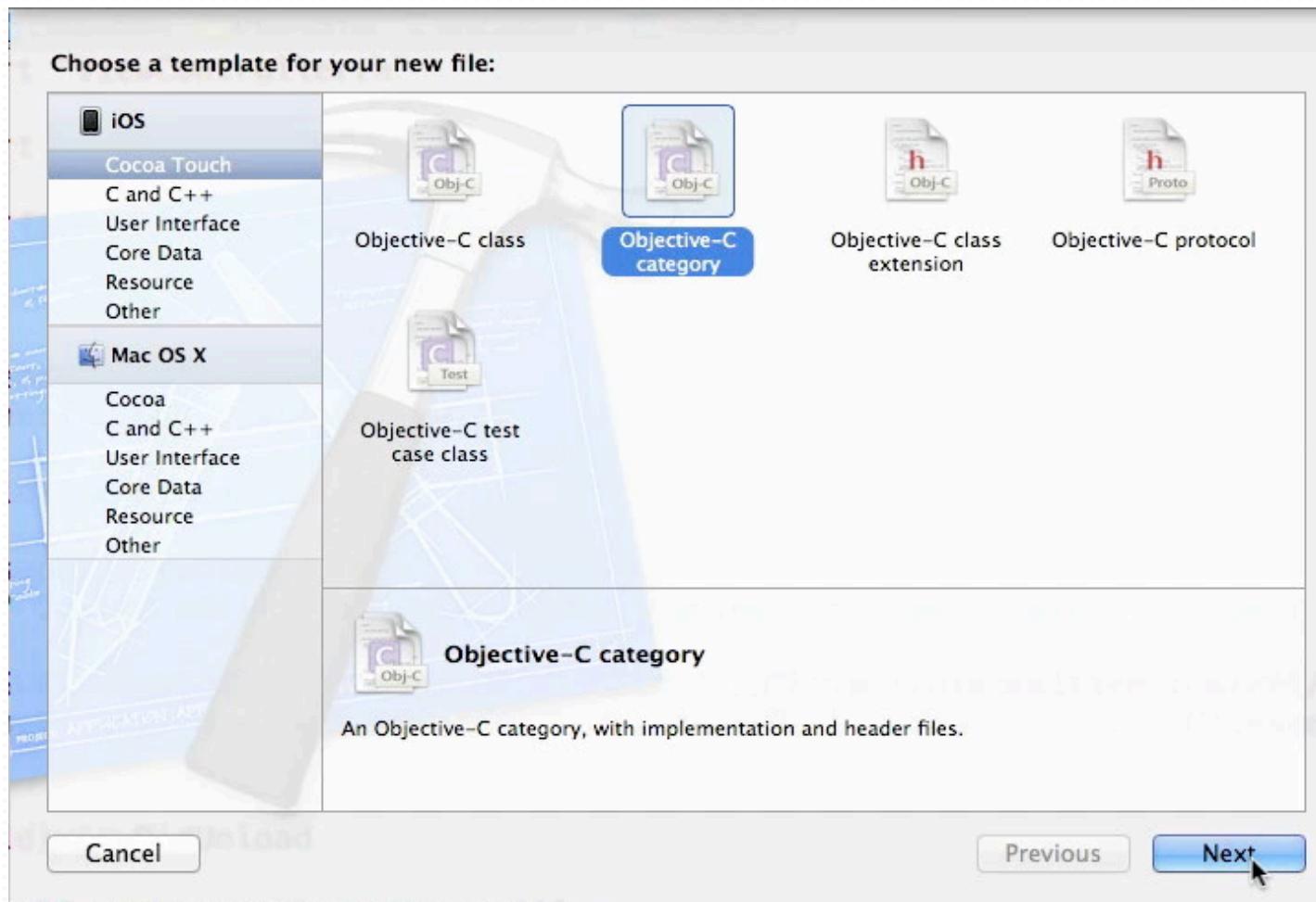
-

Category Example



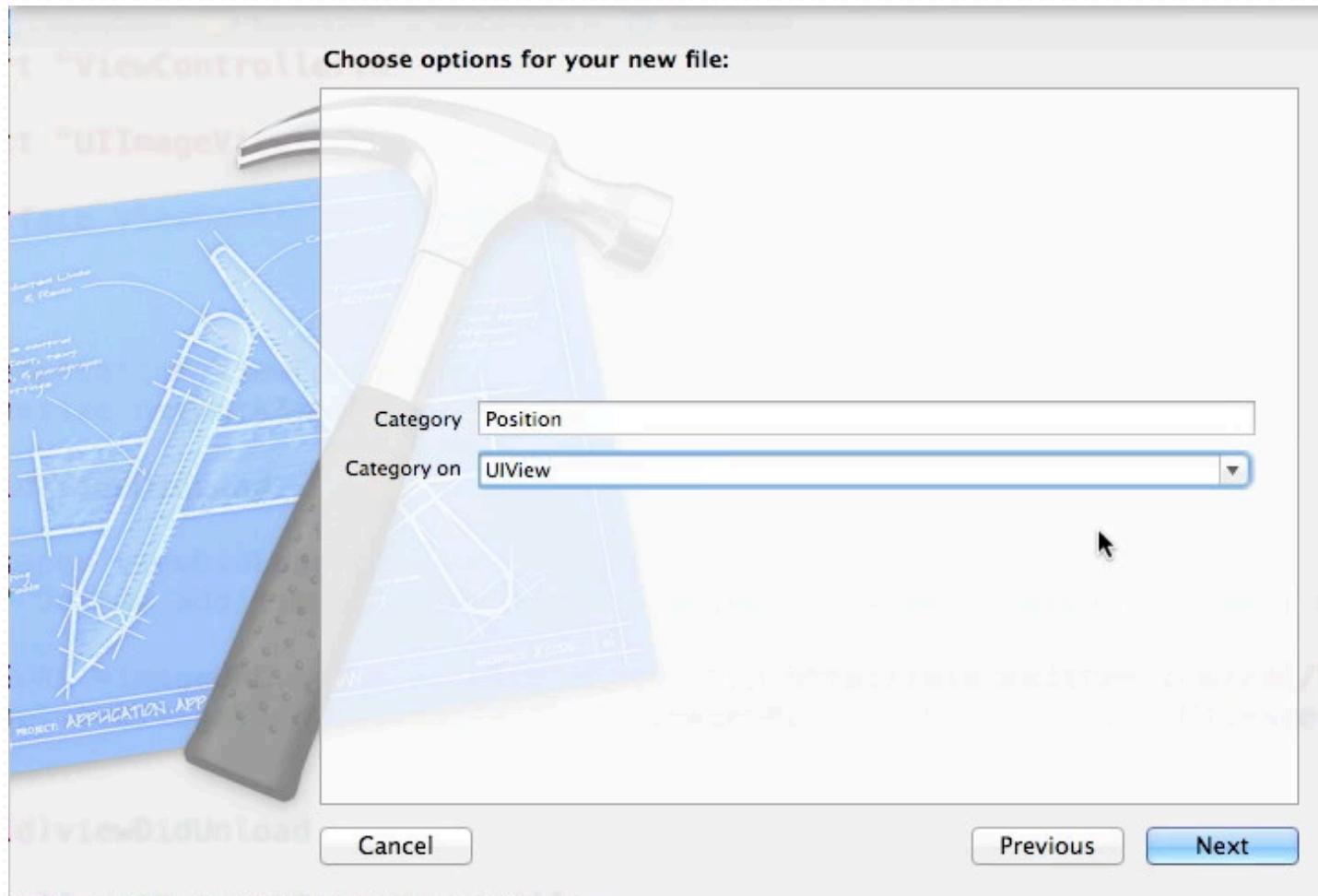
Create a new file in project

Category Example



Choose Category file

Category Example



Name the category Position on class UIView

Category Example

```
//  
//  UIView+Position.m  
//  CategoryDemo  
//  
//  Created by Freshman on 5/24/12.  
//  Copyright (c) 2012 __MyCompanyName__. All  
//  rights reserved.  
  
#import "UIView+Position.h"  
  
@implementation UIView (Position)  
  
- (void)setX:(float)x andY:(float)y  
{  
    CGRect frame = self.frame;  
    frame.origin.x = x;  
    frame.origin.y = y;  
    self.frame = frame;  
}  
  
@end
```

```
//  
//  UIView+Position.h  
//  CategoryDemo  
//  
//  Created by Freshman on 5/24/12.  
//  Copyright (c) 2012 __MyCompanyName__. All  
//  rights reserved.  
  
#import <UIKit/UIKit.h>  
  
@interface UIView (Position)  
  
// set the position of the UIView to (x,y)  
- (void)setX:(float)x andY:(float)y;  
  
@end
```

Put above code in the category files

Category Example

```
#import "UIImageView+AFNetworking.h"  
#import "UIView+Position.h"
```

Back to ViewController class and import our category

Category Example

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    NSURL * imageURL = [NSURL URLWithString:@"http://placekitten.com/200/300"];
    [networkImageView setImageWithURL:imageURL placeholderImage:[UIImage imageNamed:@"loading.png"]];

    [networkImageView setX:0 andY:0];
}
```

Add our defined UIView position method

Category Example

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    NSURL * imageURL = [NSURL URLWithString:@"http://placekitten.com/200/300"];
    [networkImageView setImageWithURL:imageURL placeholderImage:[UIImage imageNamed:@"loading.png"]];

    [networkImageView setX:0 andY:0];

    [UIView animateWithDuration:5 animations:^{
        [networkImageView setX:320 andY:480];
    }];
}
```

It's boring. Let's animate it with animate method

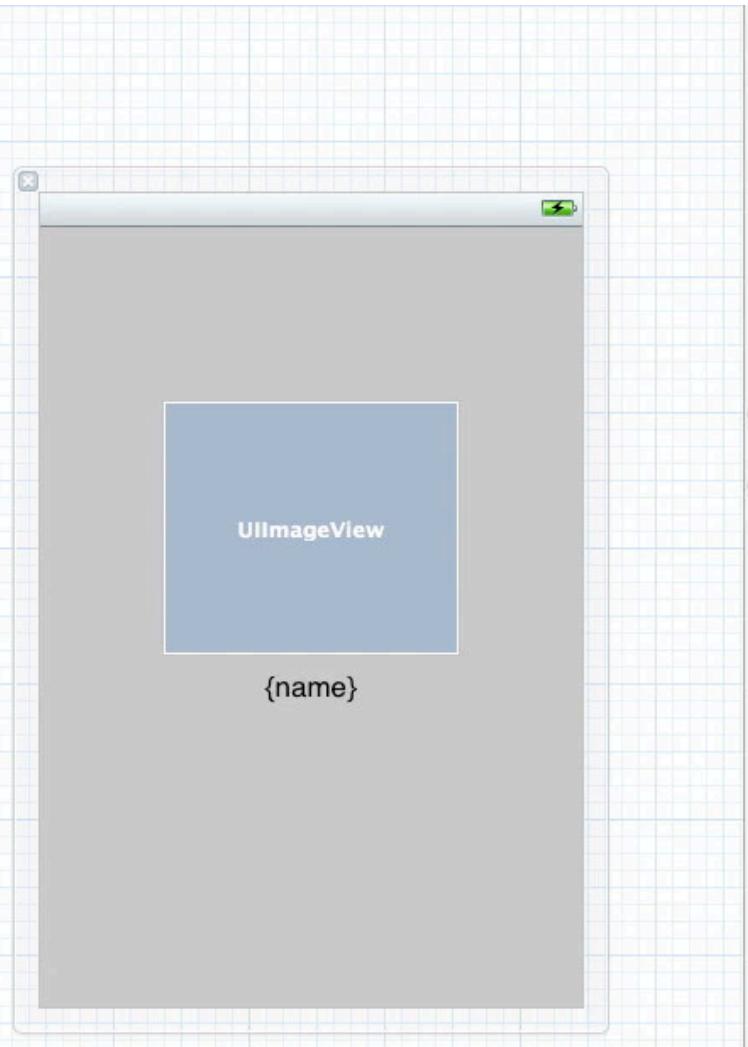
Category Example



The result with the cat moving from top left to bottom right

Querying Facebook Profile Picture

Loading FB Profile Pic



```
//  
//  ViewController.h  
//  AFNetworkTest  
//  
//  Created by Freshman on 5/24/12.  
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved  
  
#import <UIKit/UIKit.h>  
  
@interface ViewController : UIViewController  
@property (retain, nonatomic) IBOutlet UIImageView *  
    networkImageView;  
@property (retain, nonatomic) IBOutlet UILabel *nameLabel;  
  
@end
```

Add a label in interface

Loading FB Profile Pic

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    NSURL * imageURL = [NSURL URLWithString:@"http://graph.facebook.com/makzan/picture?type=large"];
    [networkImageView setImageWithURL:imageURL placeholderImage:[UIImage imageNamed:@"loading.png"]];

}
```

Change the image URL to your profile pic. Done.

Facebook Graph API

Loading JSON

```
[super viewDidLoad];
// Do any additional setup after loading the view, typically from a nib.

NSURL *imageURL = [NSURL URLWithString:@"http://graph.facebook.com/makzan/picture?type=large"];
[networkImageView setImageWithURL:imageURL placeholderImage:[UIImage imageNamed:@"loading.png"]];

NSURL *url = [NSURL URLWithString:@"http://graph.facebook.com/makzan"];
NSURLRequest *request = [NSURLRequest requestWithURL:url];
AFJSONRequestOperation *operation = [AFJSONRequestOperation JSONRequestOperationWithRequest:request success:^(NSURLRequest *
request, NSHTTPURLResponse *response, id JSON) {
    NSLog(@"Result: %@", JSON);
    nameLabel.text = [JSON objectForKey:@"name"];
} failure:^(NSURLRequest *request, NSHTTPURLResponse *response, NSError *error, id JSON) {
    // Handle error here.
}];
[operation start];
```

We use the AFJSONRequestOperation to query API
in JSON format.

Accessing NSDictionary

Accessing NSDictionary

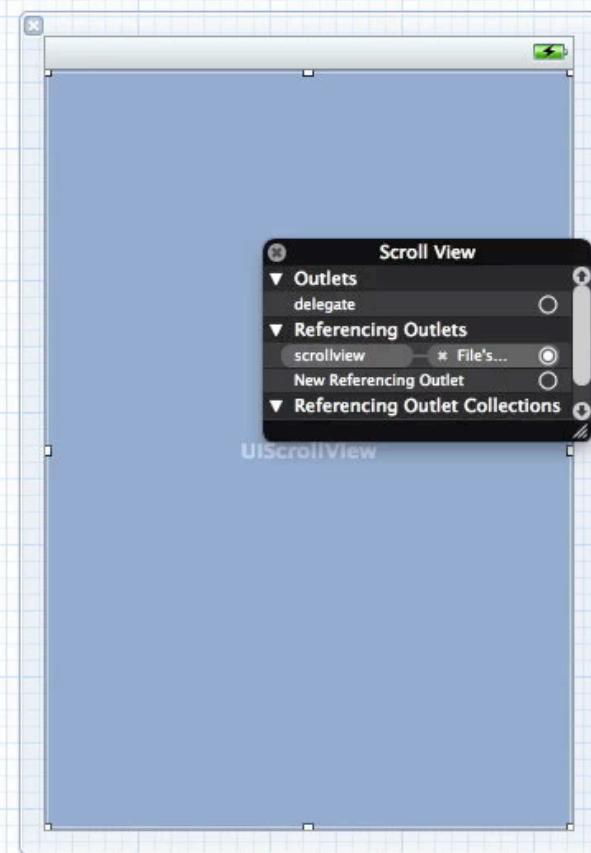
```
// dict is a NSDictionary instance.  
[dict objectForKey:@”key”];
```

Accessing NSDictionary

```
// dict is a NSDictionary instance.  
[dict setValue:@"John" forKey:@"name"];
```

Using UIScrollView

UIScrollView



```
// ViewController.h
// AFNetworkTest
//
// Created by Freshman on 5/24/12.
// Copyright (c) 2012 __MyCompanyName__. All rights reserved.

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

@property (retain, nonatomic) IBOutlet UIScrollView *scrollview;

@end
```

Drag a UIScrollView into view and put it full screen.
Name it scrollview

UIScrollView

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    UIImageView *image = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"320x460.png"]];
    [scrollview addSubview:image];

    [scrollview setContentSize:CGSizeMake(320*2, 460)];
}
```

We add an image view inside scroll view

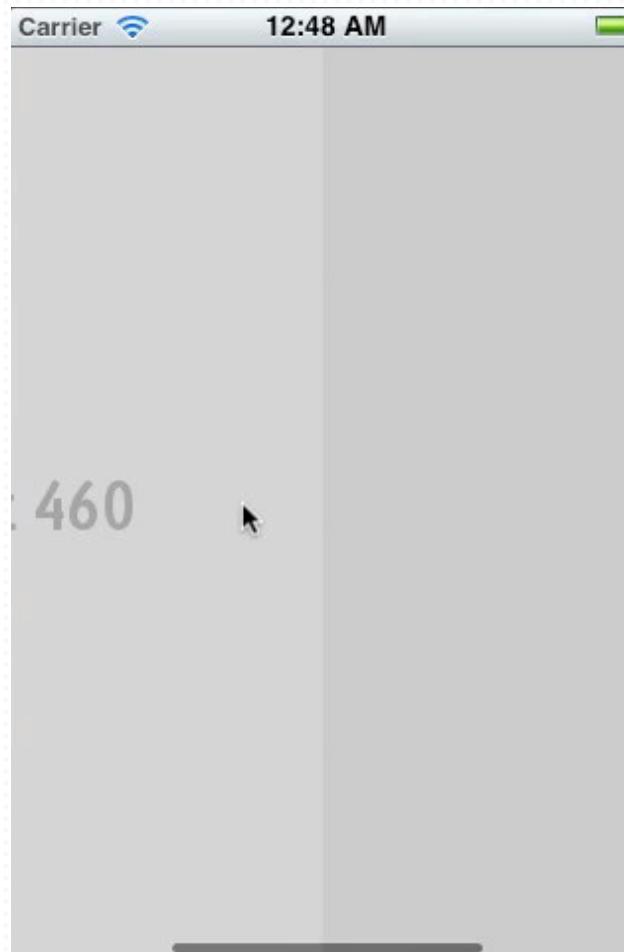
Subview

- Every UIView contains sub views.
- Sub views are also UIView.
- When we move the parent views, all sub views follows.
- You can think UIView is a container of other UIViews.

Adding and Removing Subview

- [view addSubview: view2];
- [view2 removeFromSuperview];
- [view insertSubview:view2 atIndex:2];
- [view insertSubview:view2 aboveSubview: view3];
- [view insertSubview:view2 belowSubview: view3];

UIScrollView



Test it in iPhone and we get the image inside scroll view

UIScrollView

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    for (int i=0; i<4; i++)
    {
        UIImageView *image = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"320x460.png"]];
        [scrollview addSubview:image];

        [image setX:320 * i andY:0];
    }

    [scrollview setContentSize:CGSizeMake(320*4, 460)];
}
```

Let's try add 4 images inside the scroll view

UIScrollView

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    for (int i=0; i<4; i++)
    {
        UIImageView *image = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"320x460.png"]];
        [scrollview addSubview:image];

        [image setX:320 * i andY:0];
    }

    [scrollview setContentSize:CGSizeMake(320*4, 460)];
    scrollview.pagingEnabled = YES;
}
```

We can enable page by page scrolling with pagingEnabled

Using Interface Builder with UIScrollView

UIScrollView in Interface Builder

- Easy to accidentally put views inside UIScrollView
- Need to arrange views carefully

Exercise

Combining all we learned so far

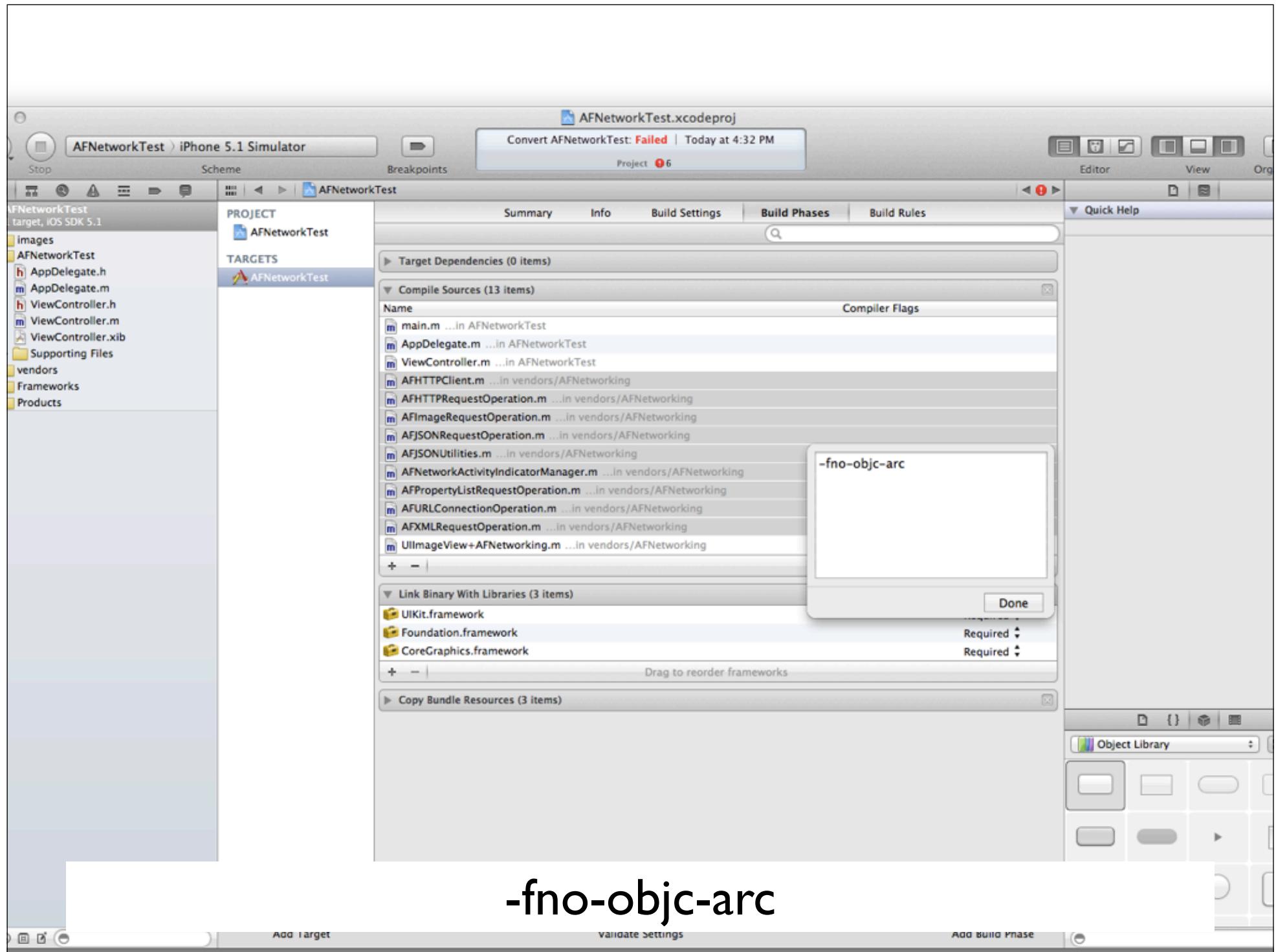


Query Facebook Photos API and display them in scroll view

Avoiding files compiling with ARC

No ARC Flag

1. Project Setting
2. Build Phrase
3. Compile Sources
4. Apply -fno-objc-arc



-fno-objc-arc

