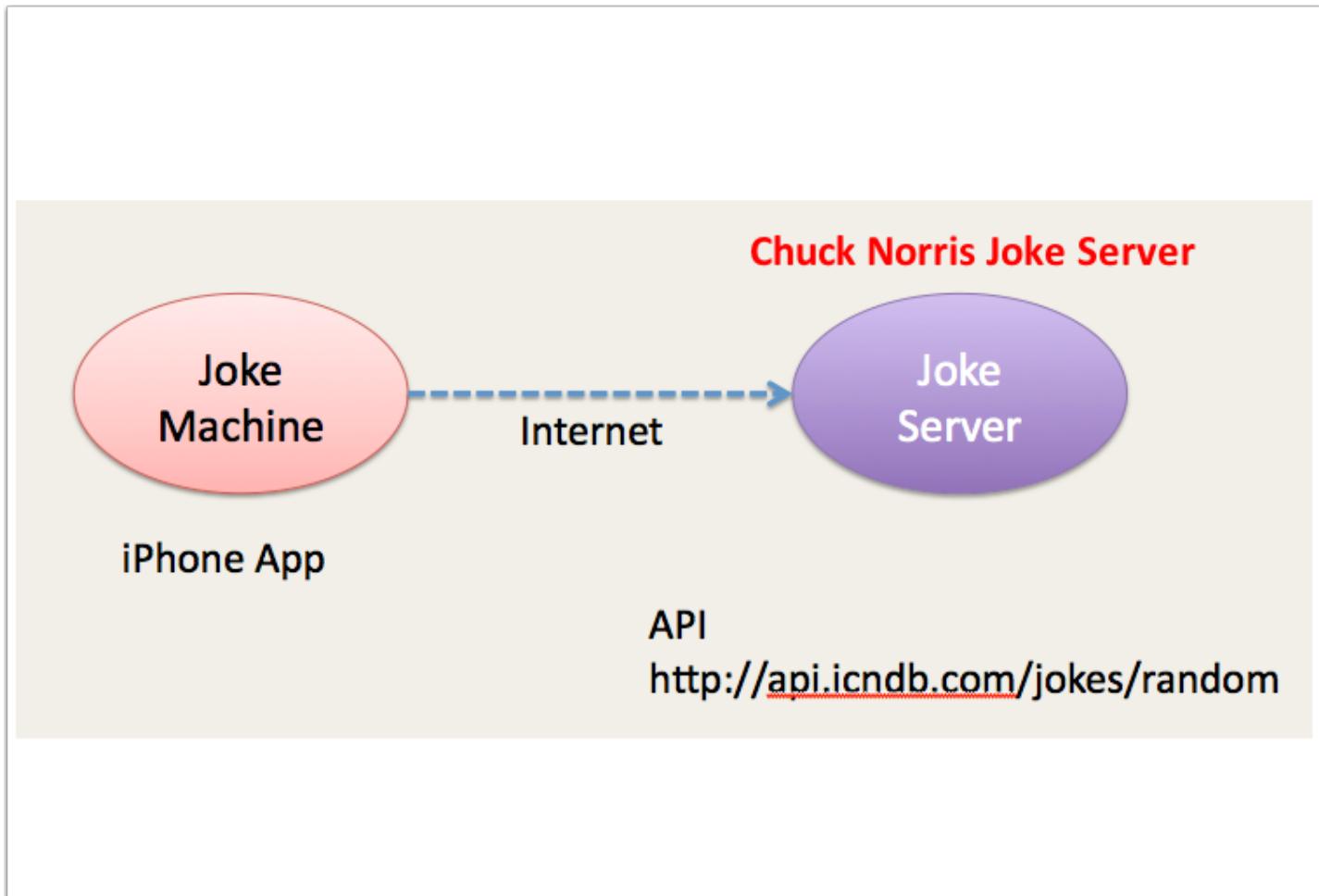


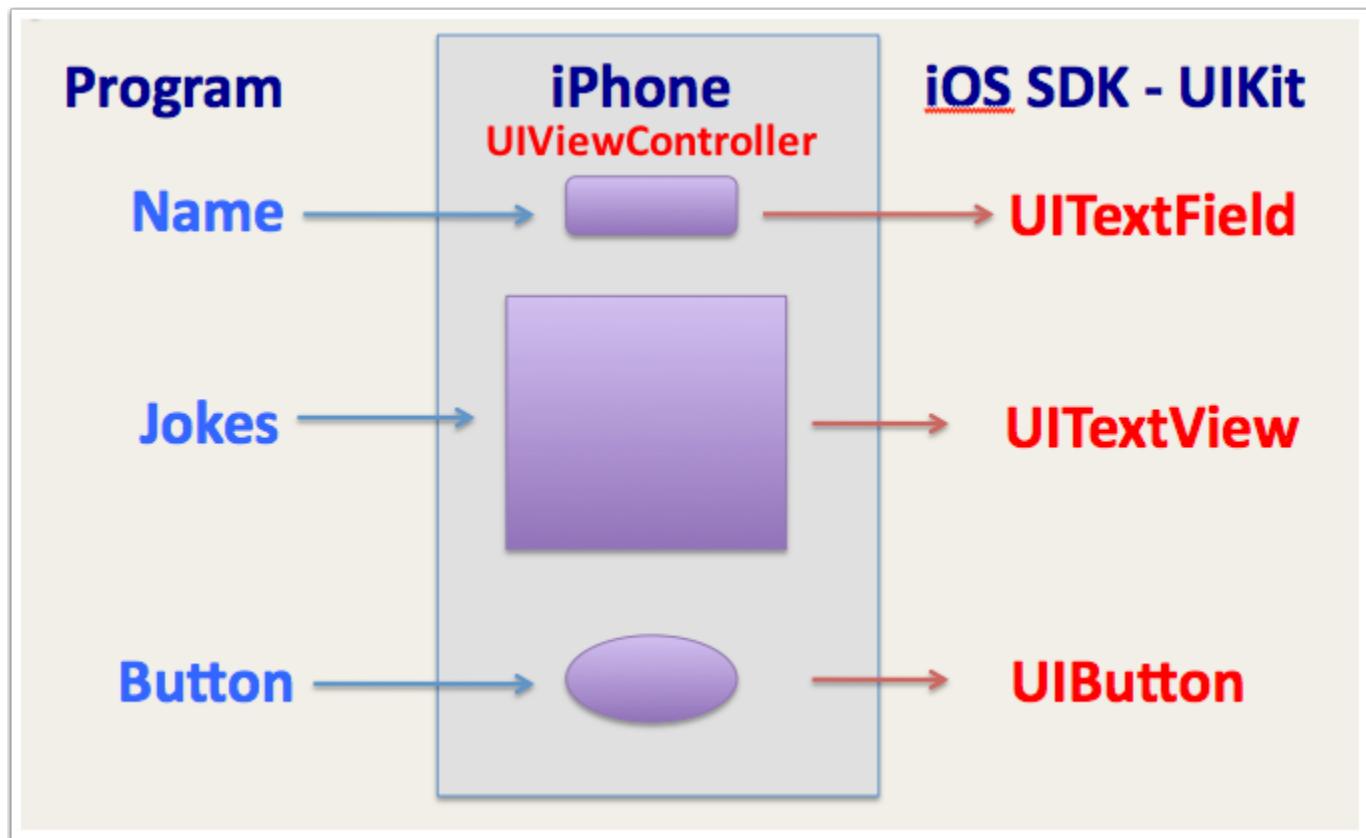
What we are building



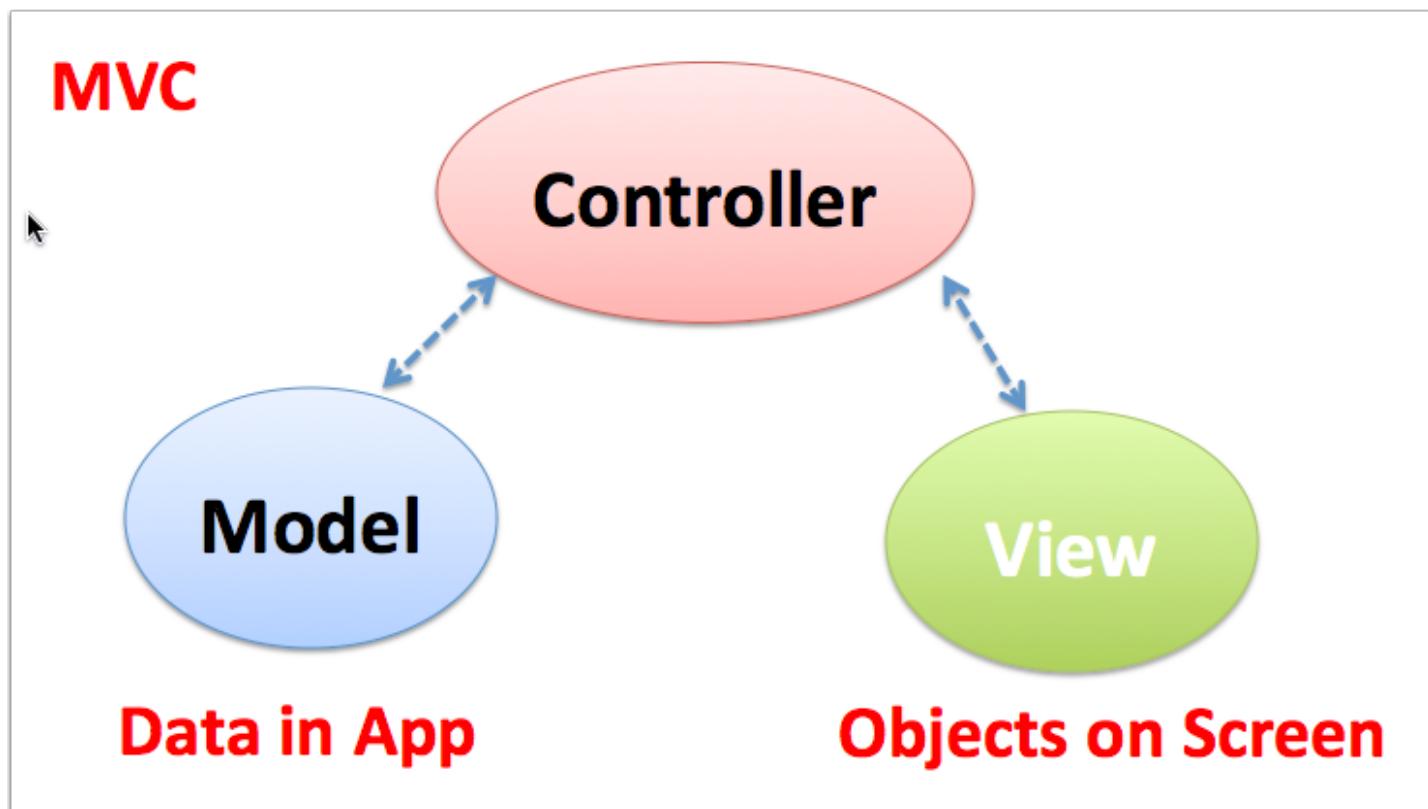
Initial Screen Shot



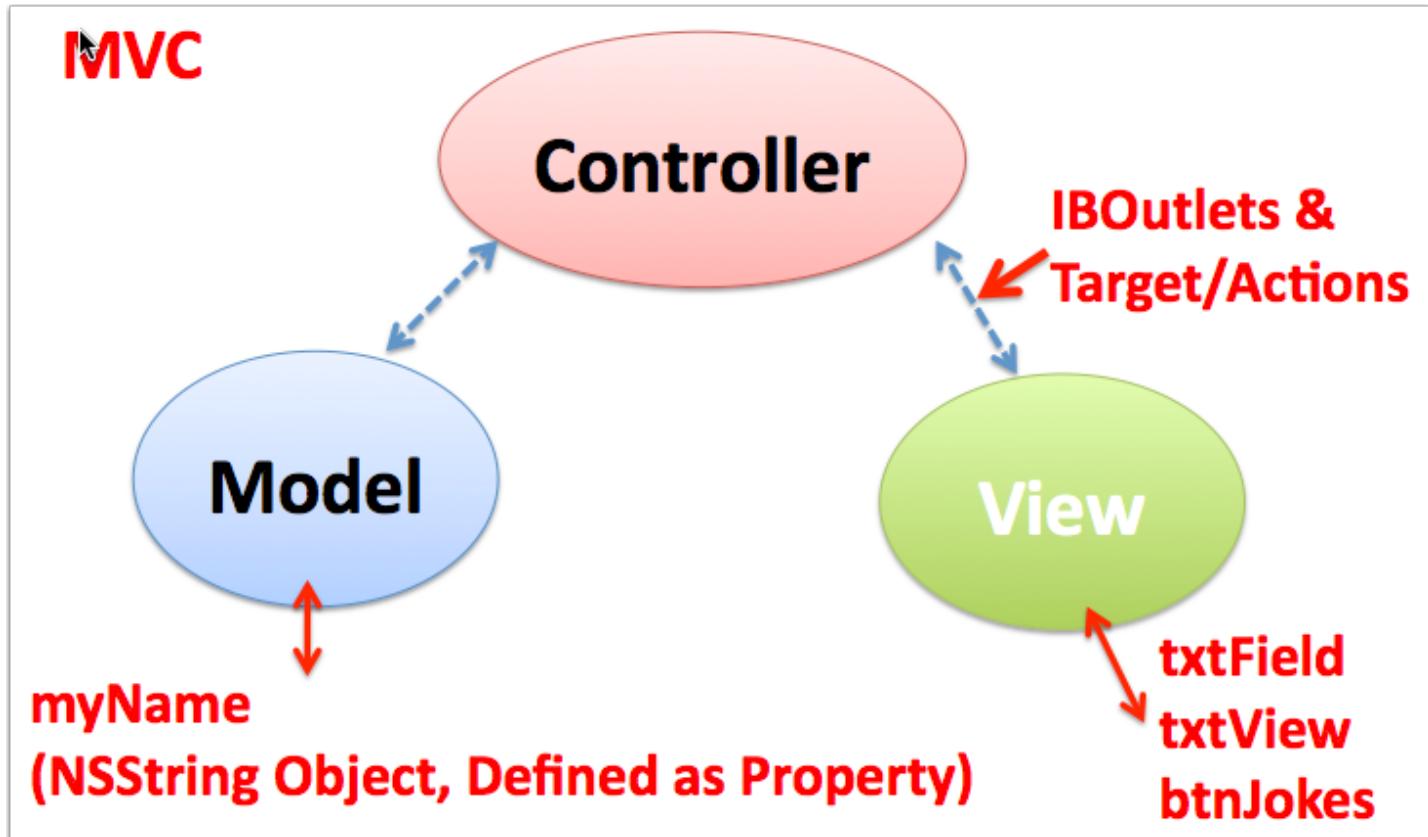
Components Required



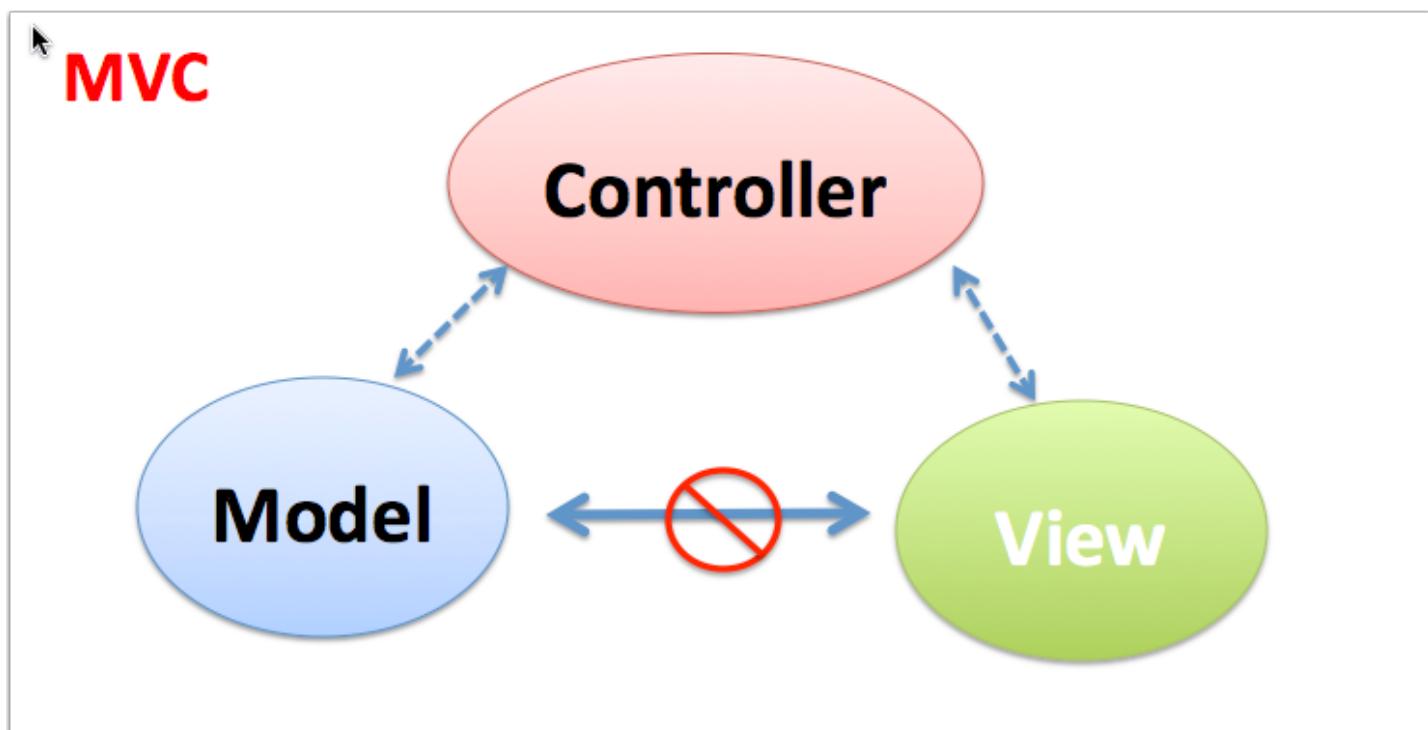
iOS Programming Paradigm - Two Defaults Controller and Storyboard



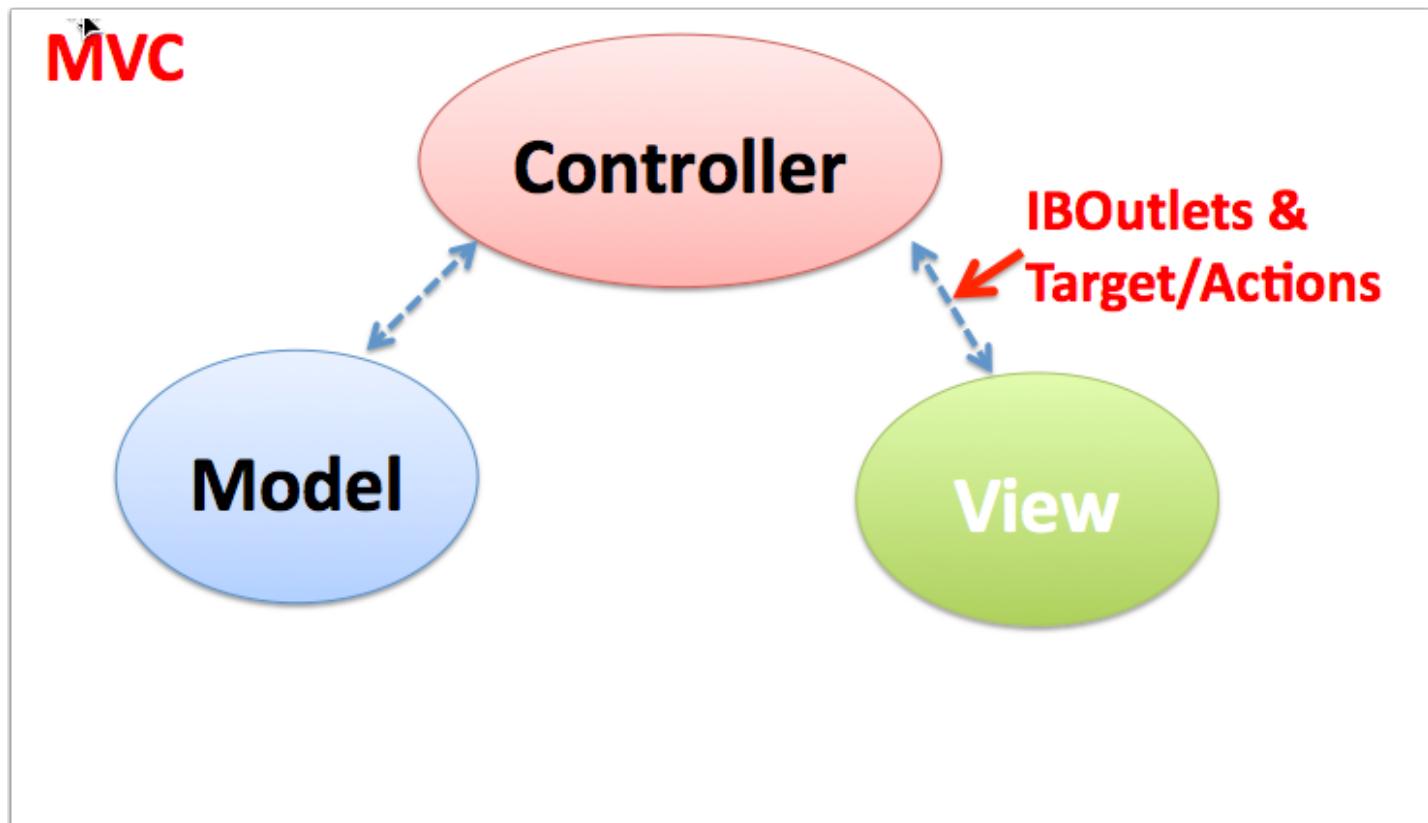
iOS Objects



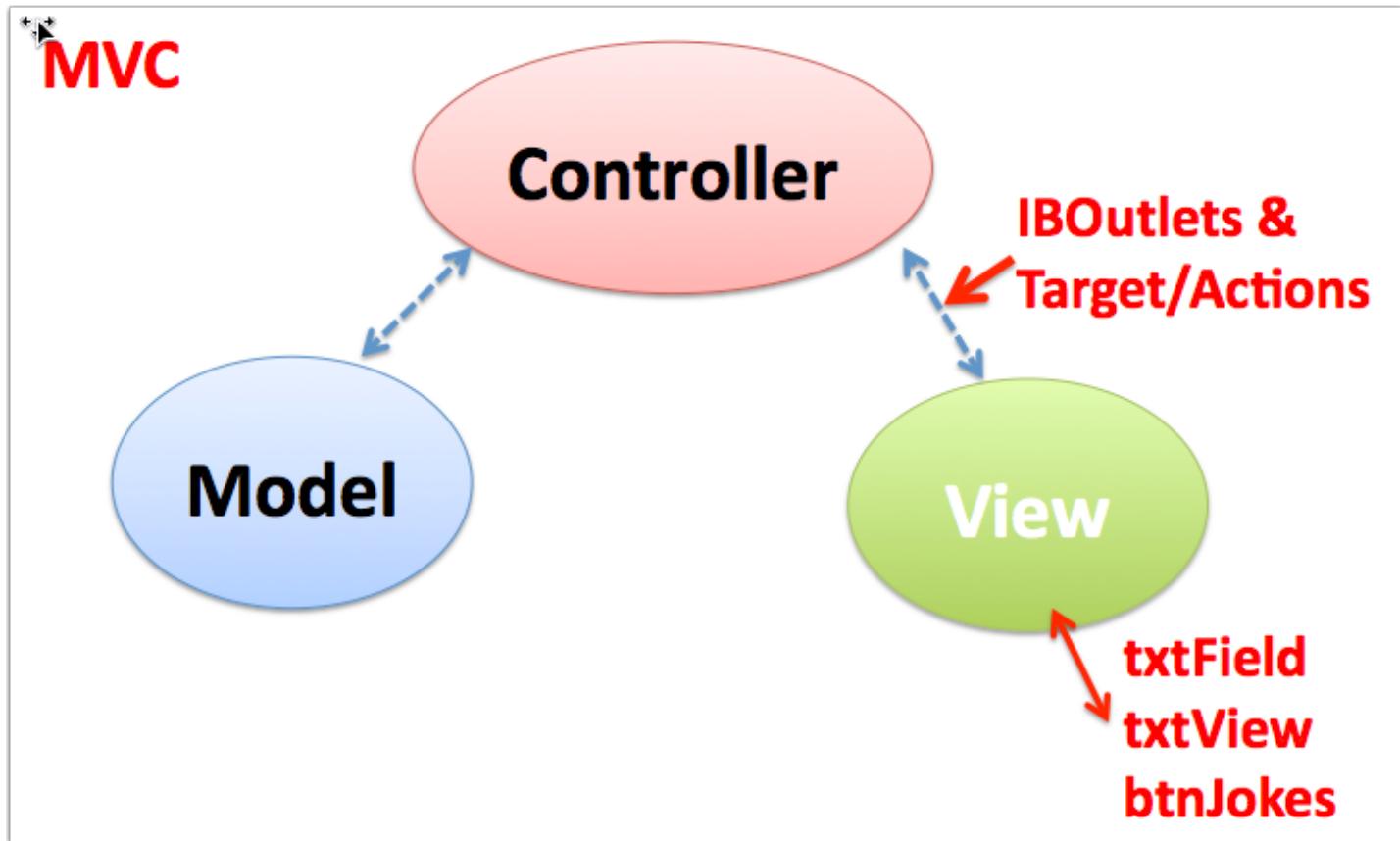
Views are Reusable (No dependency on Model)



How do Views Communicate with Controller?

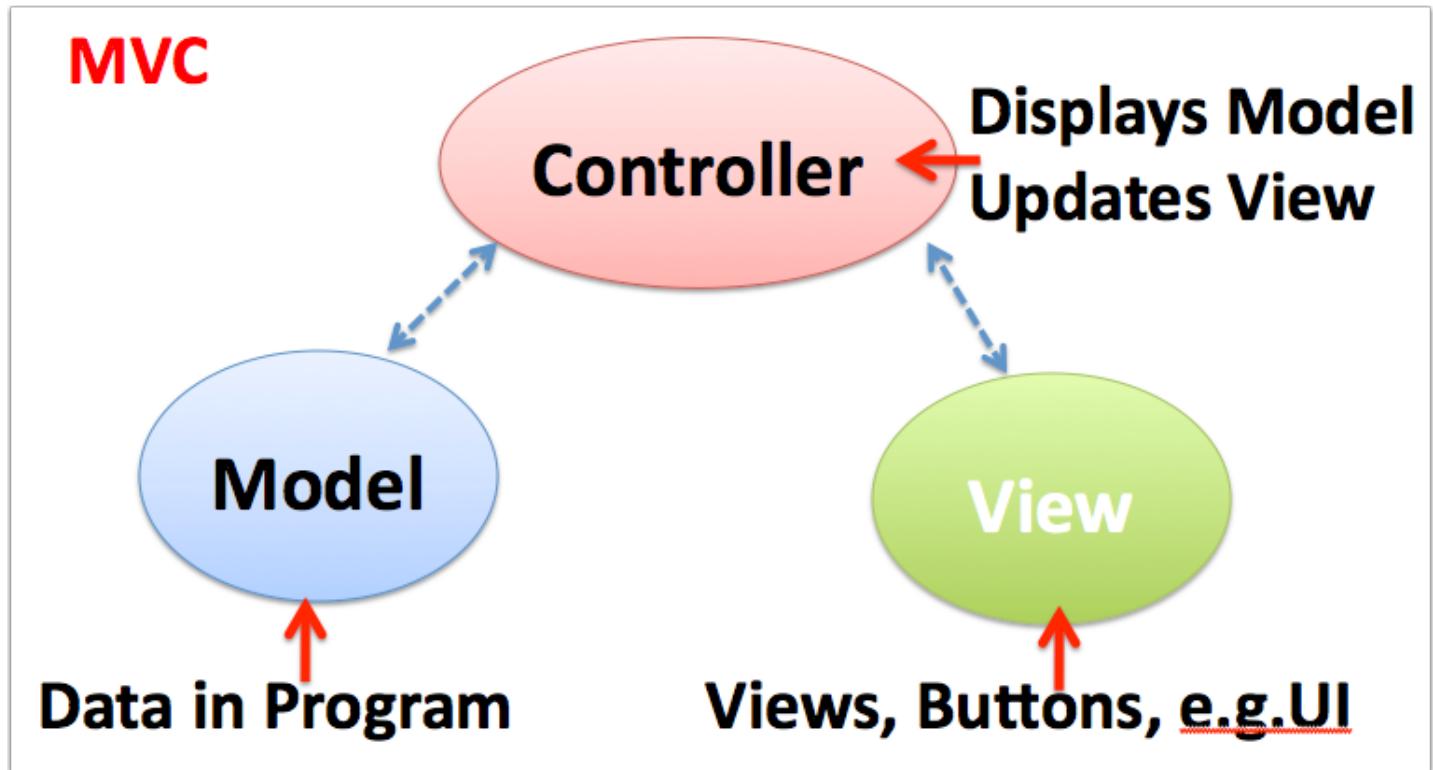


Define IBOutlets with Actions



MVC Complete

Xcode Gives you Two of these! Storyboard and Controller!



What's in the Program (Code, Storyboard, UI Components)

Model: myName Variable

View: Single View Application

Controller: MDNJViewController

Internet: NSURLConnection Class

UI: Buttons, TextField, TextView

Defining Model (Variables)

 **Variables hold values for your “Model”, Examples:**

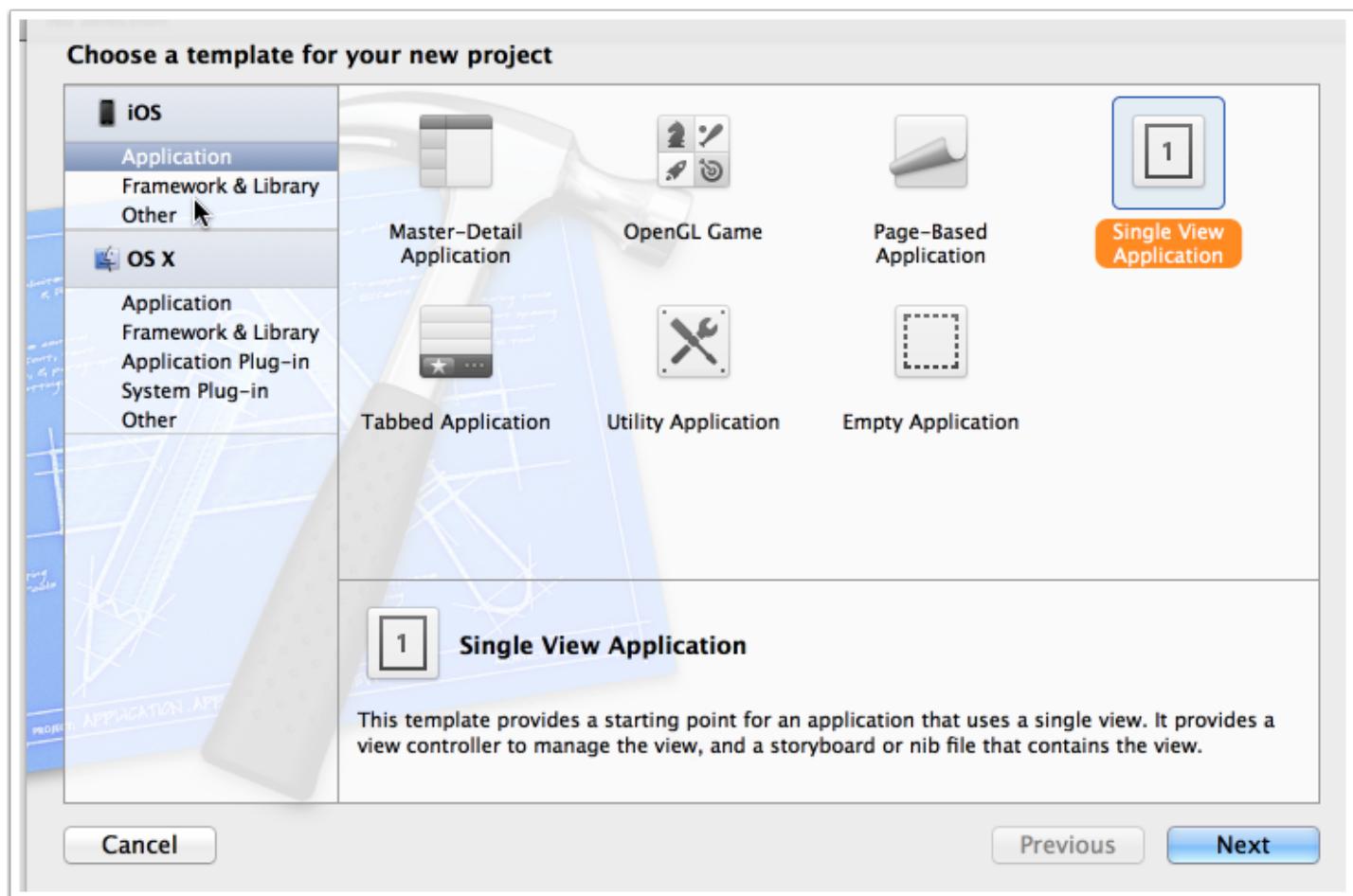
```
int Age = 25;  
NSError *error = nil;  
NSString *myName;  
UITextField *txtName;  
UITextView *txtViewJokes;
```

Defining Properties

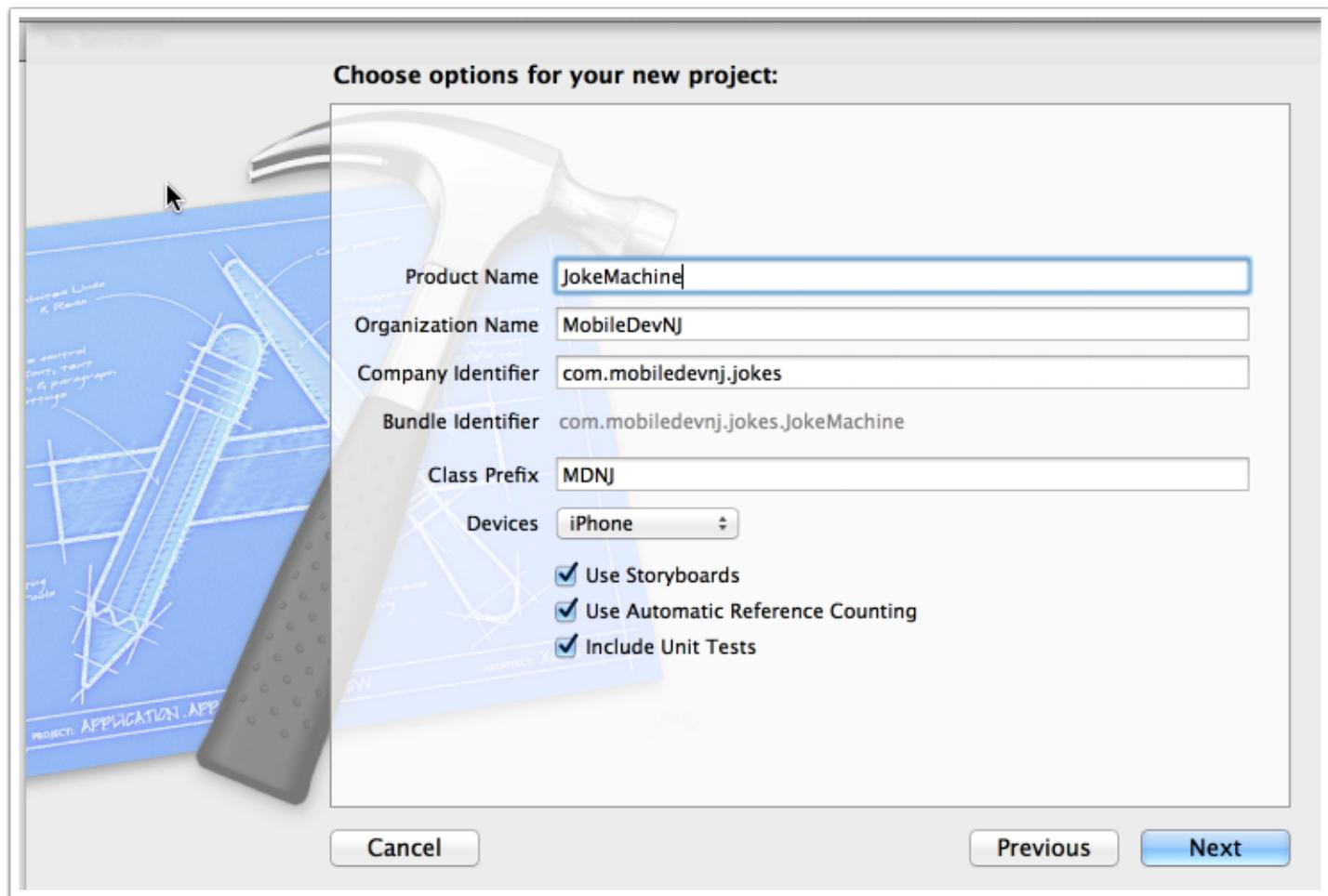
 **Some Variables are defined using @Property Keyword, Examples:**

```
@property (weak, nonatomic) UITextField  
*txtName;  
  
@property (weak, nonatomic)  
UITextView *txtViewJokes;
```

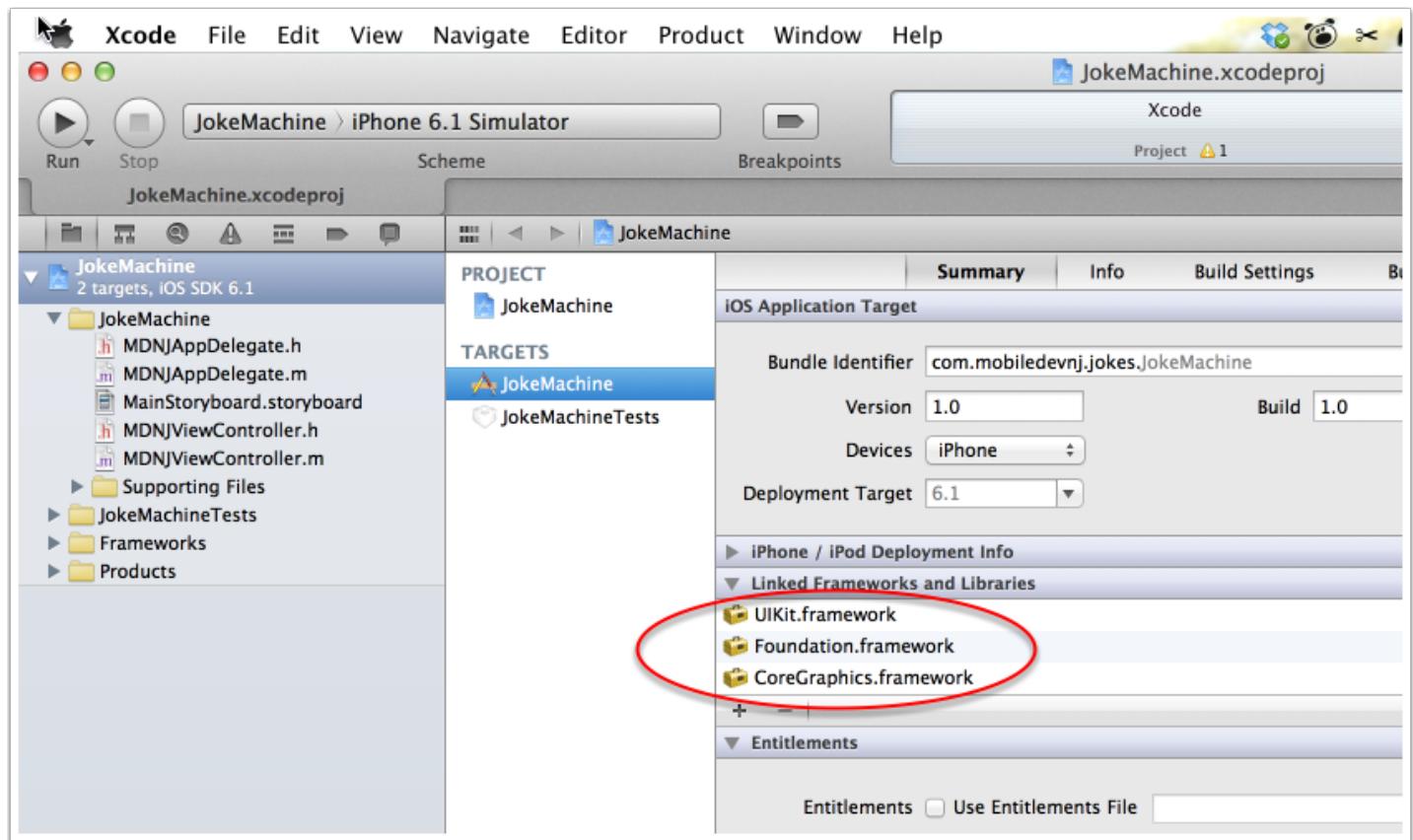
File New



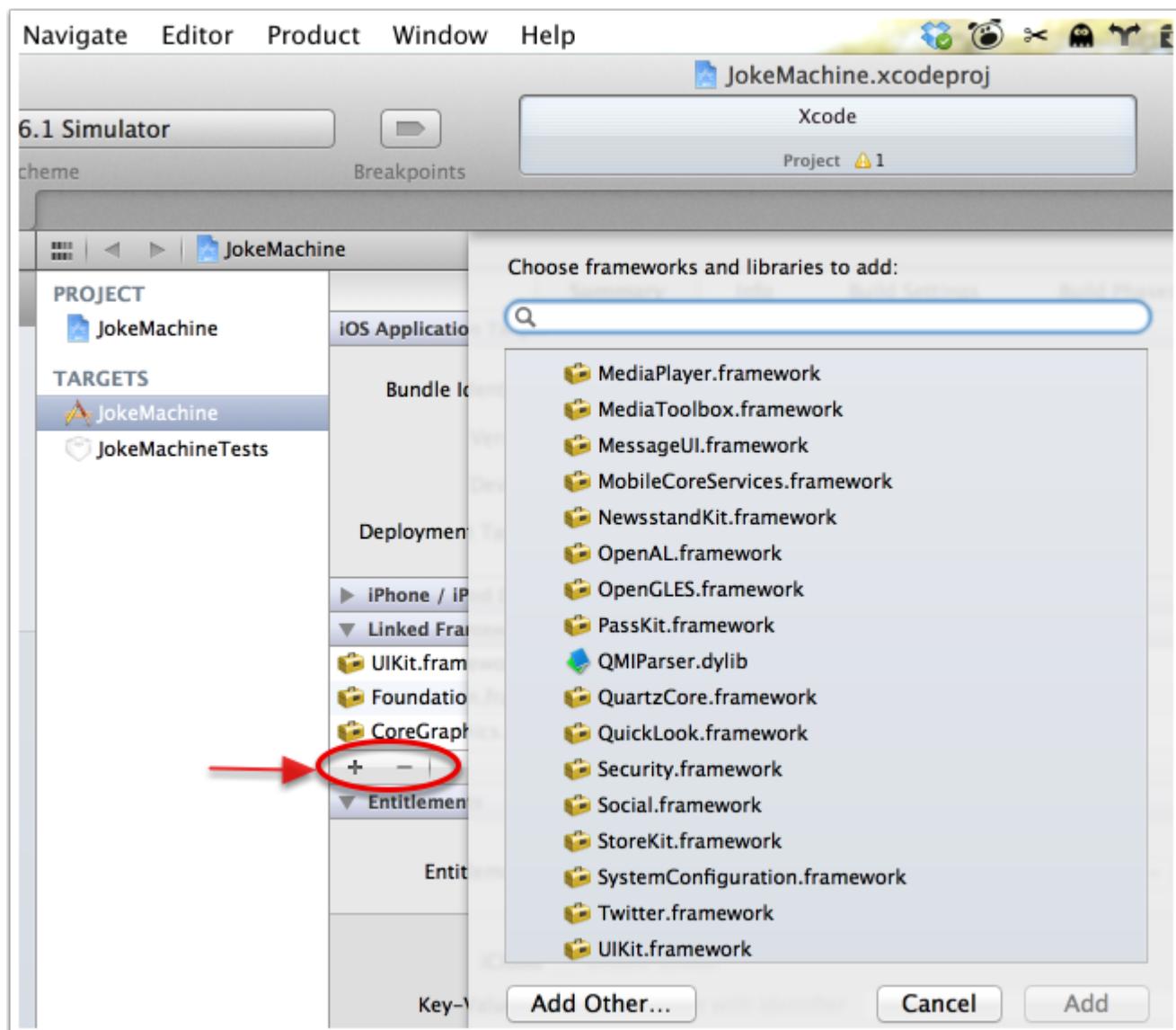
Xcode - App Name



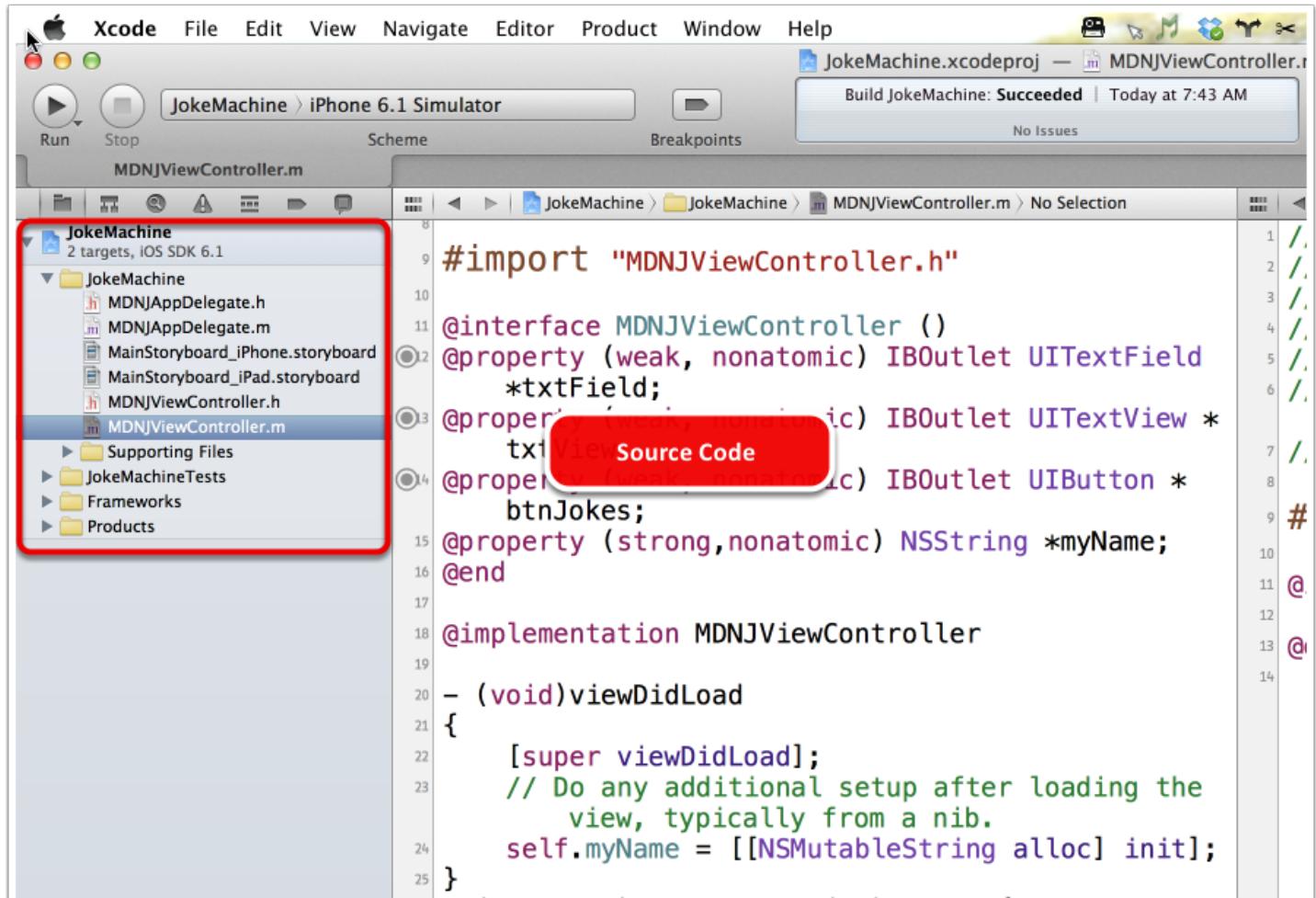
Xcode - Step 1



Xcode - Frameworks



Files Navigator



Source Code Files (Xcode)

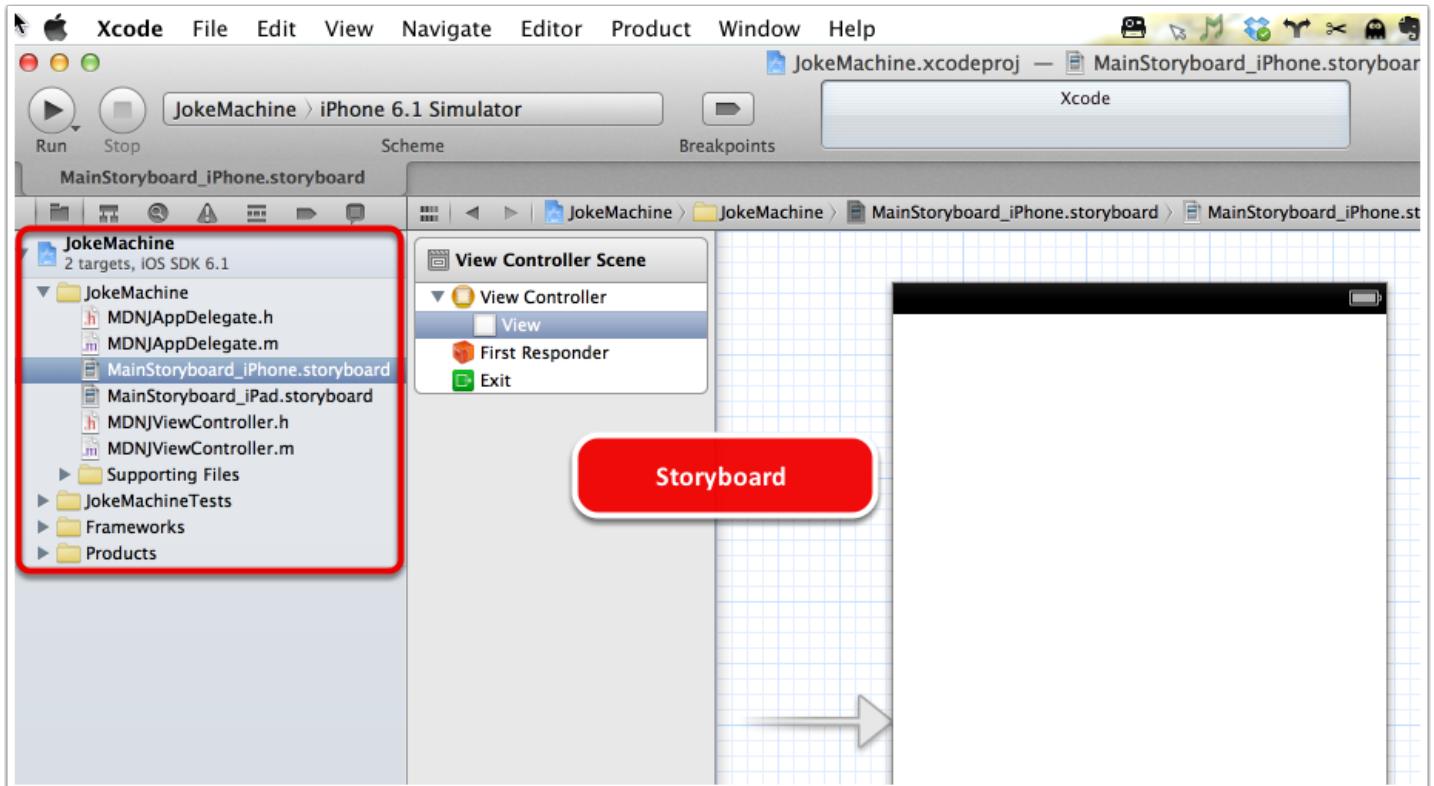
The screenshot shows the Xcode interface with the following details:

- Toolbar:** Xcode, File, Edit, View, Navigate, Editor, Product, Window, Help.
- Run/Stop Buttons:** Run, Stop.
- Scheme:** JokeMachine > iPhone 6.1 Simulator.
- Breakpoints:** Breakpoints button.
- Project Navigator:** Shows the project structure under "JokeMachine". A red box highlights the "MDNJViewController.h" and "MDNJViewController.m" files.
- Code Editor:** Displays the "MDNJViewController.m" file content. The code includes imports, interface declarations, properties, and methods like viewDidLoad and actionJokes:.
- Status Bar:** Build status: Succeeded | Today at 7:43 AM | No Issues.

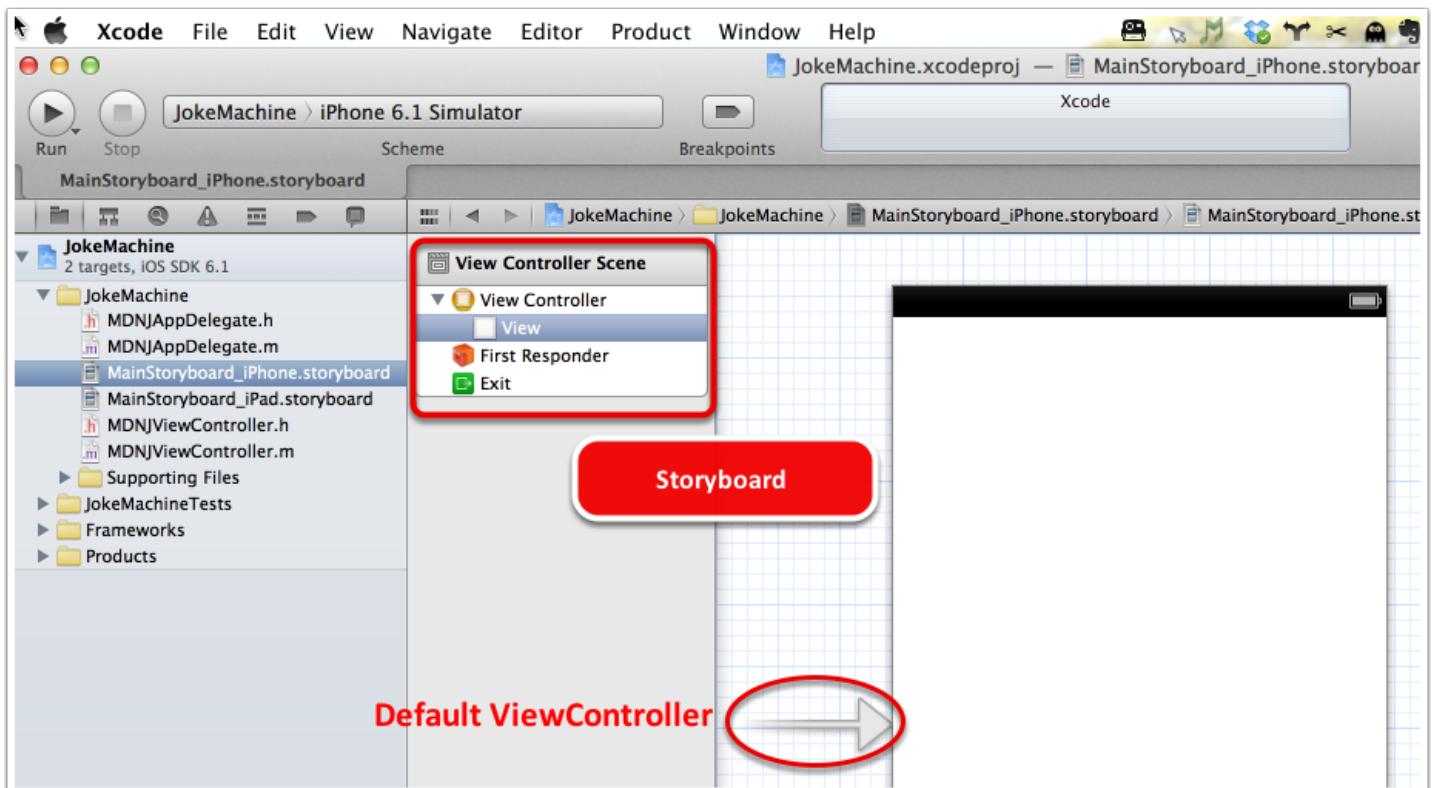
.h = Header File
.m = Objective-C Code

```
// Created by Neo on 5/6/13.  
// Copyright (c) 2013 MobileDevNJ. All rights reserved.  
  
#import "MDNJViewController.h"  
  
@interface MDNJViewController : UIViewController  
@property (weak, nonatomic) IBOutlet UITextField *txtField;  
@property (weak, nonatomic) IBOutlet UITextView *txtView;  
@property (weak, nonatomic) IBOutlet UIButton *btnJokes;  
@property (strong, nonatomic) NSString *myName;  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.  
    self.myName = [[NSMutableString alloc] init];  
}  
- (IBAction)actionJokes:(id)sender {  
    // Get Jokes from API over Internet  
}
```

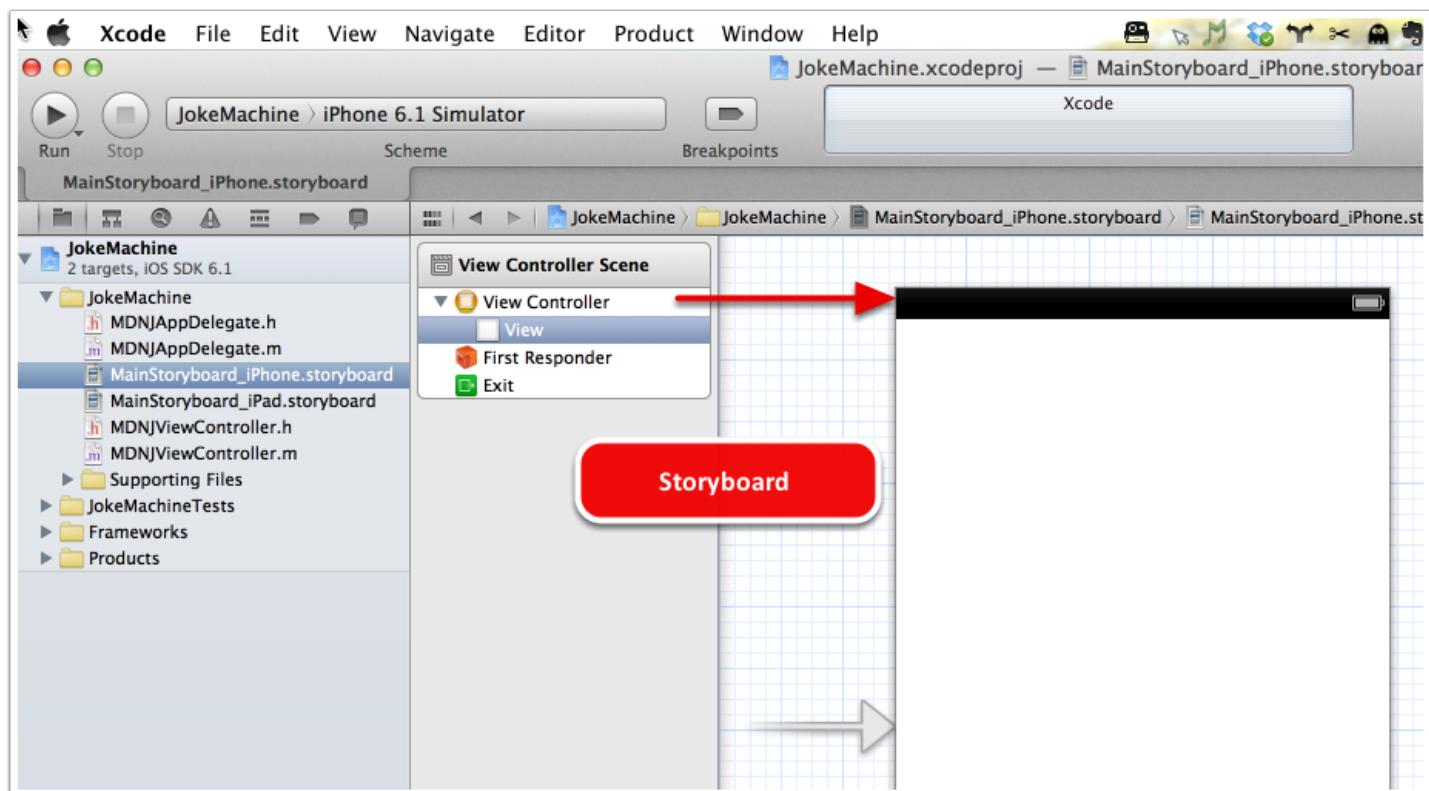
Storyboard (Xcode)



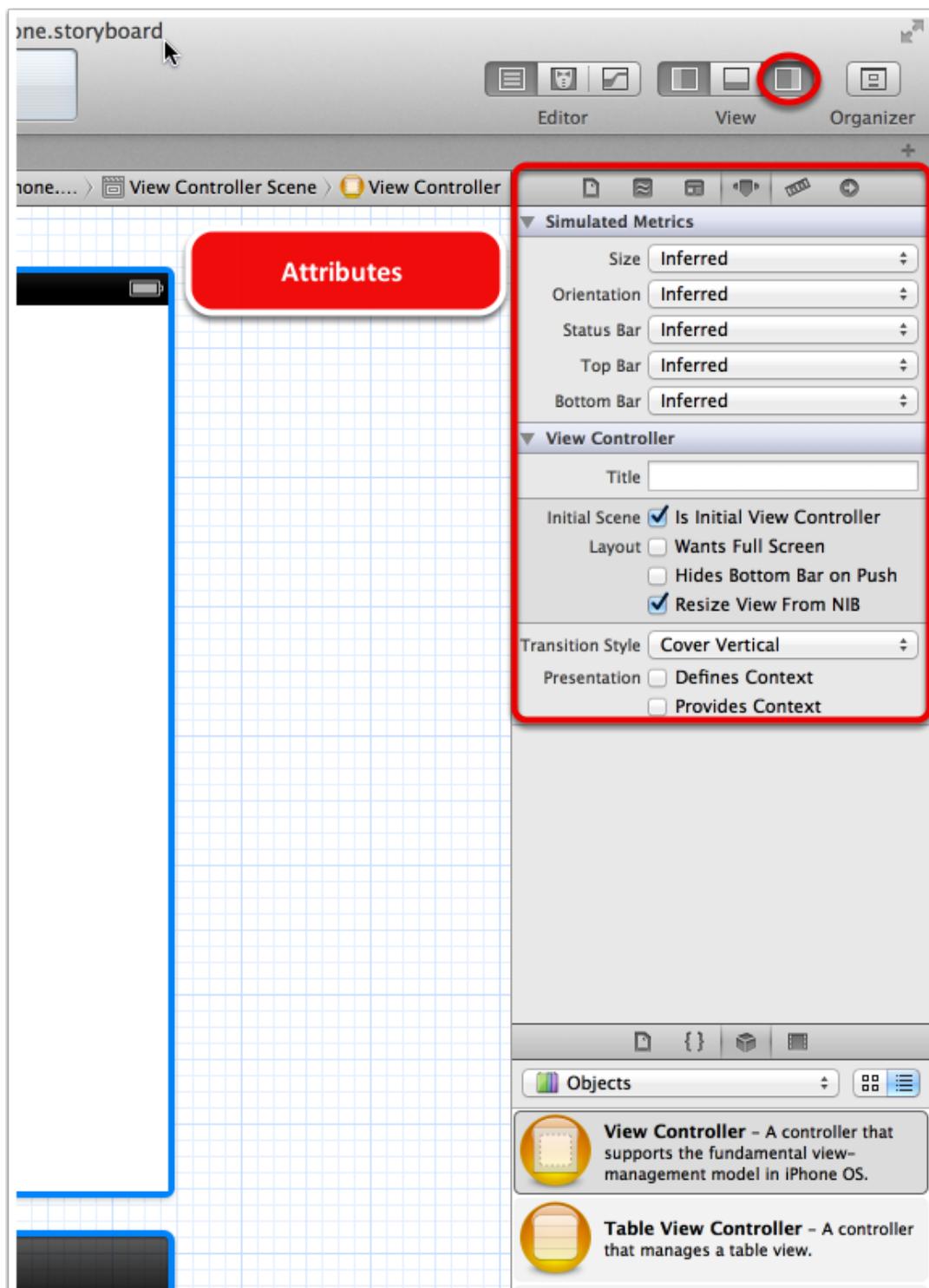
Storyboard (Xcode Objects)



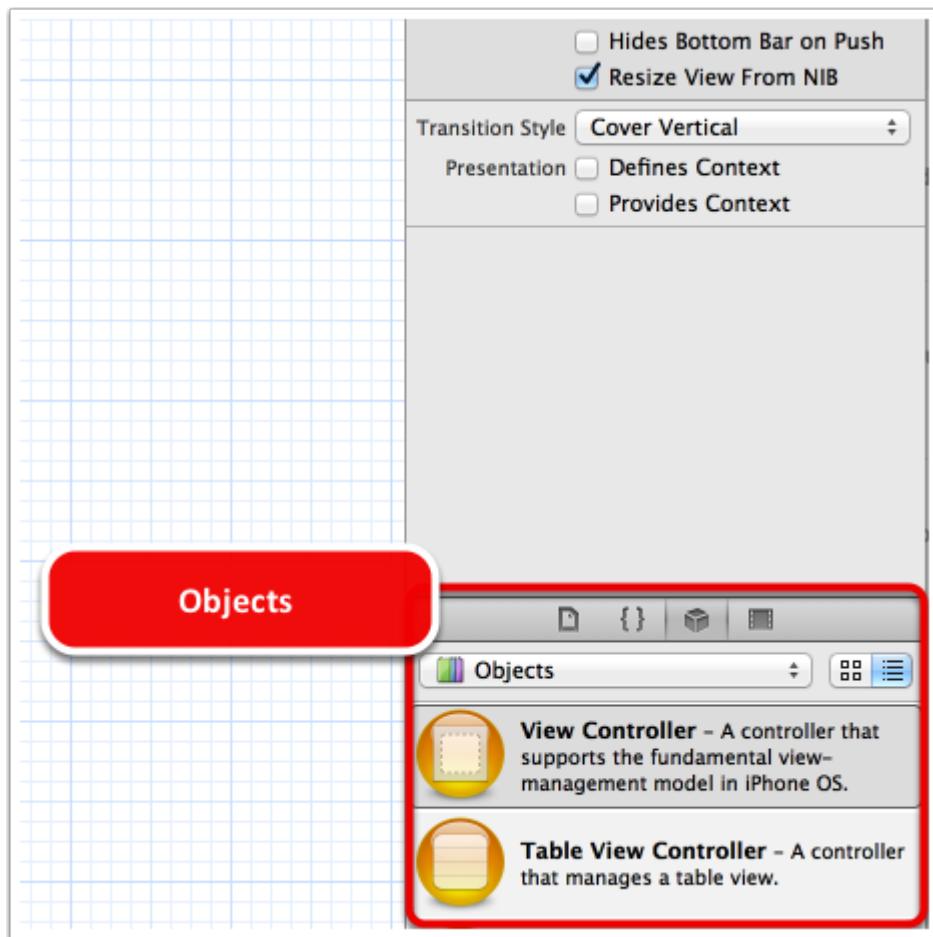
ViewController (Xcode)



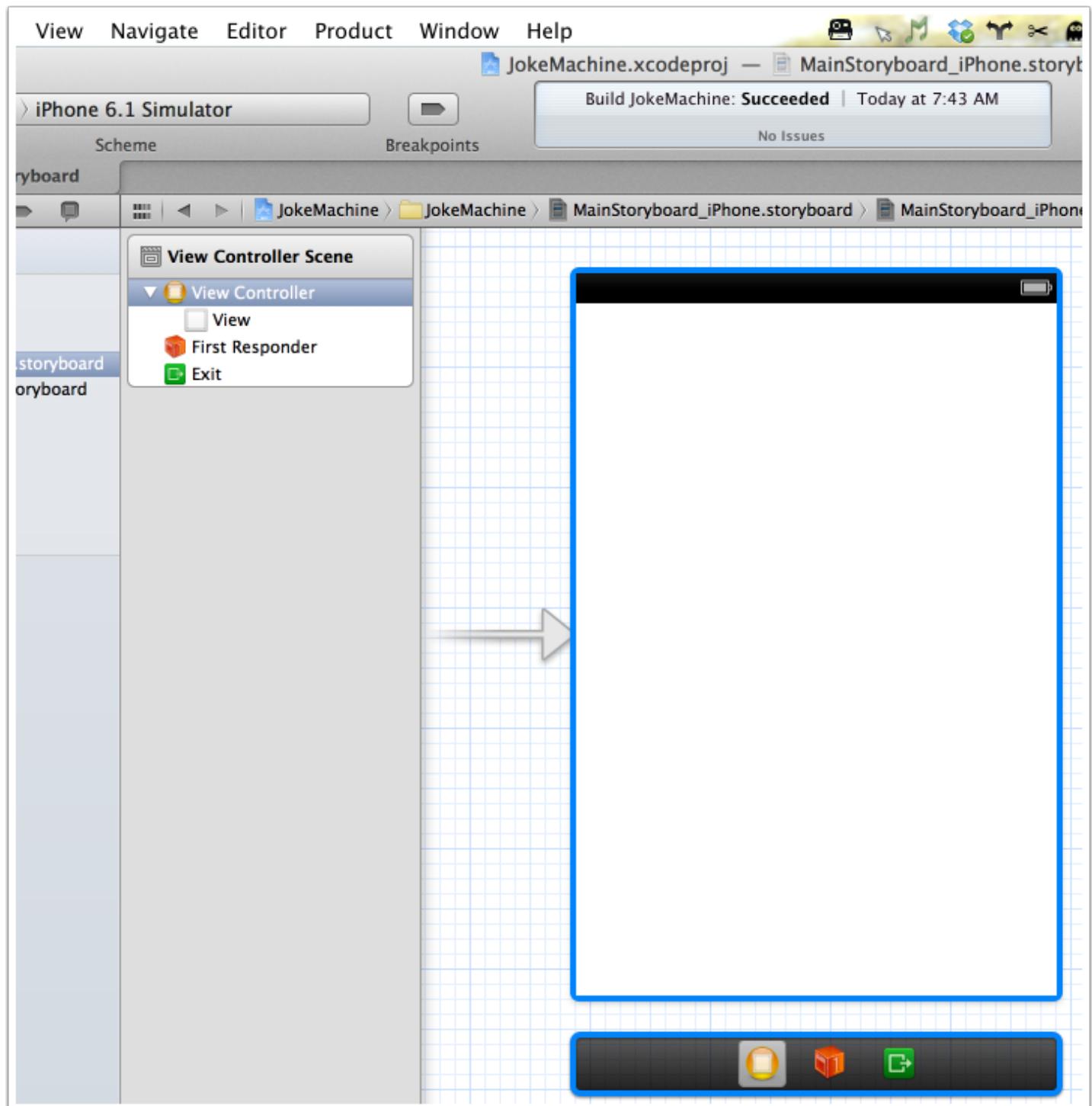
Utility Pane - Attributes



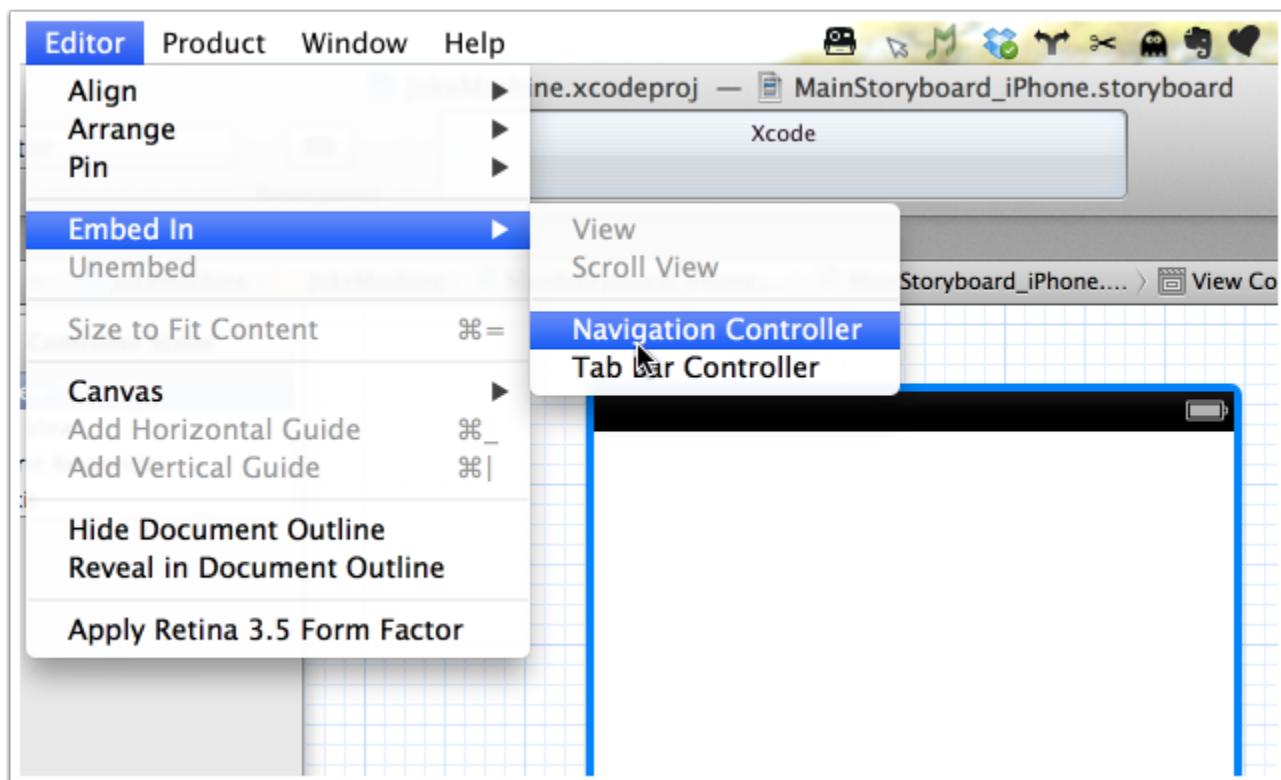
Utility Pane - UIKit Objects



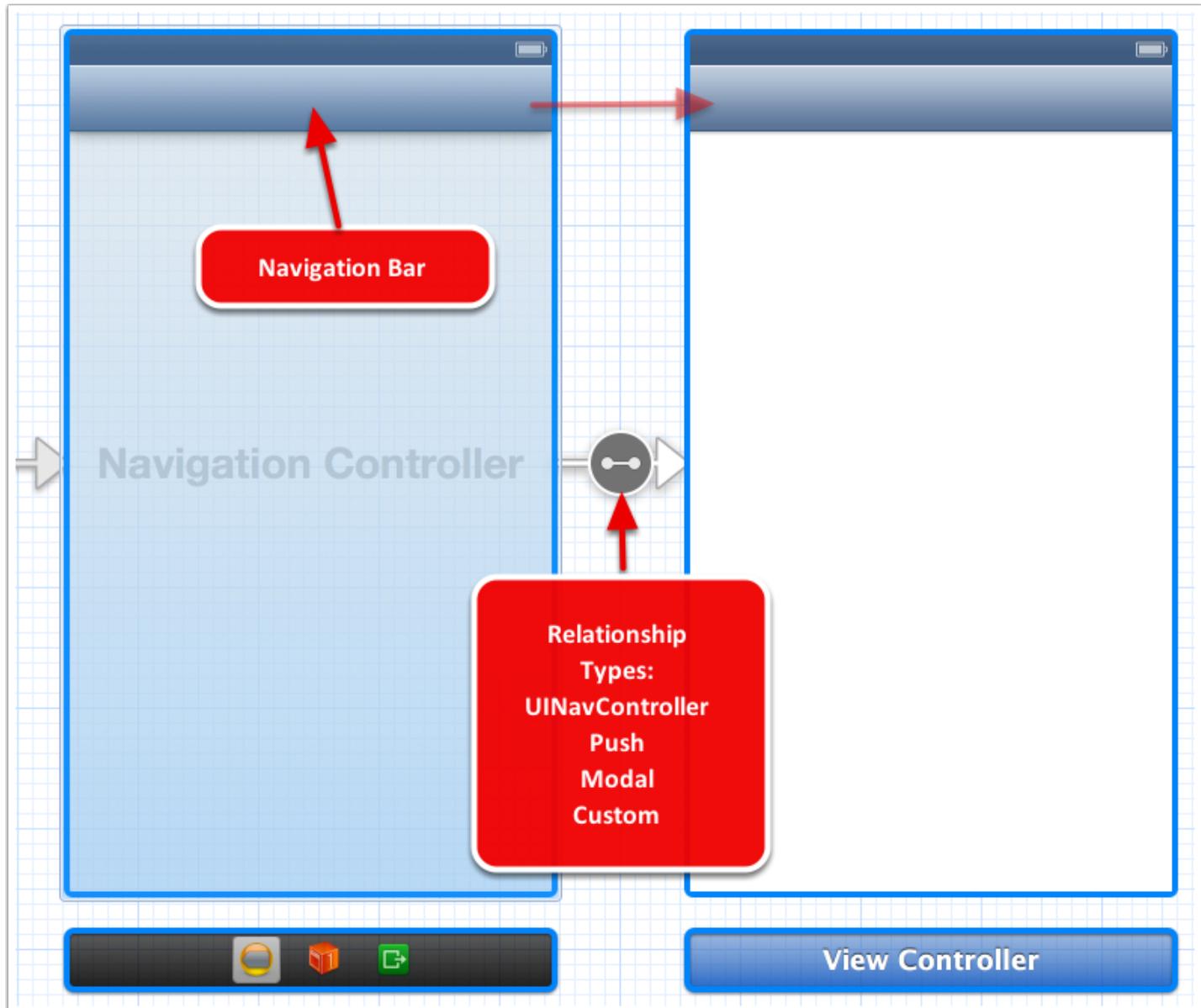
MDNJUIViewController



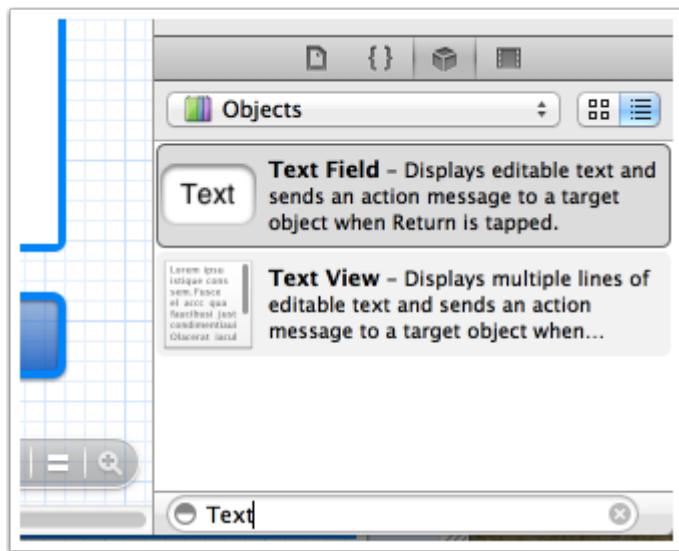
Adding UINavigationController (Navigation)



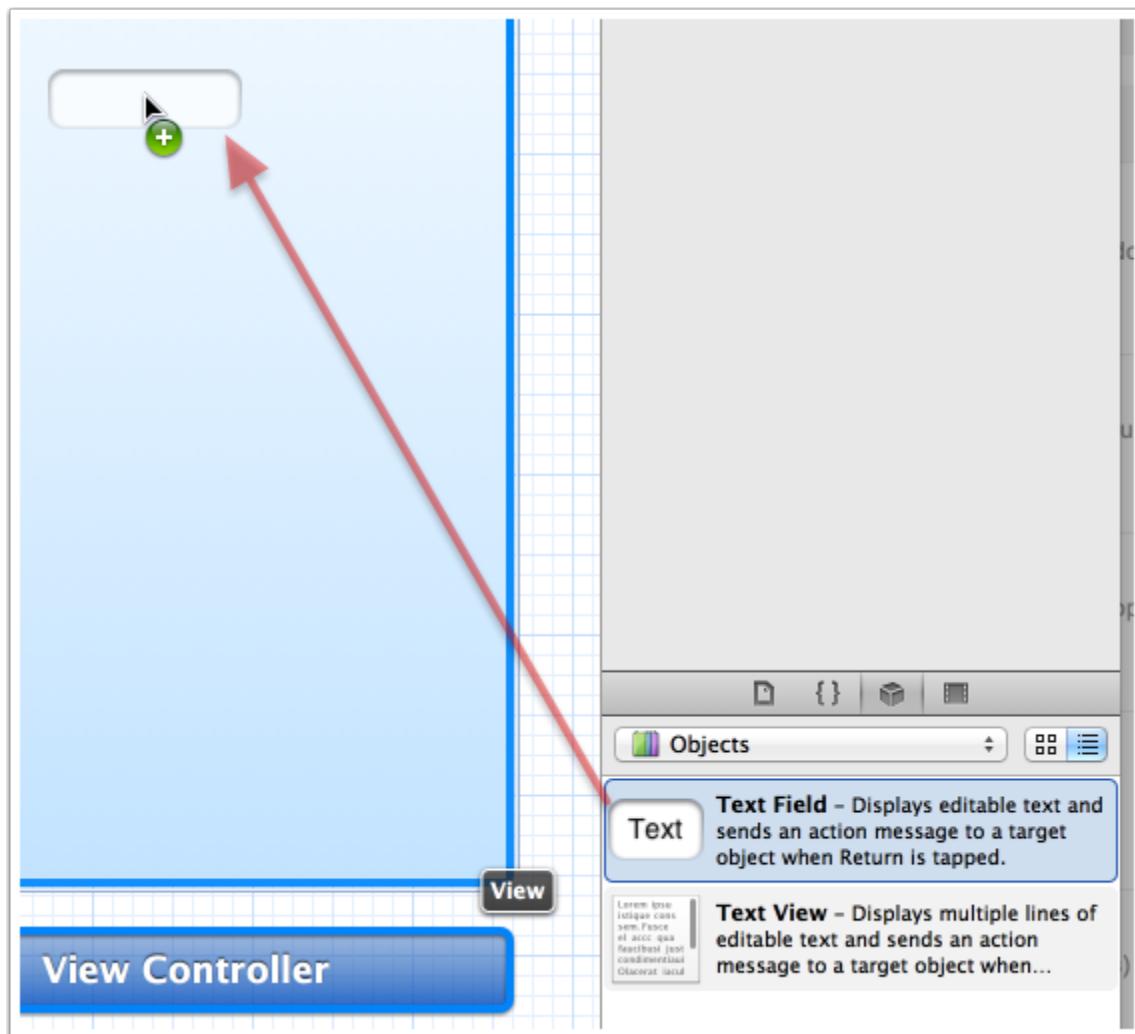
UINavigationBar



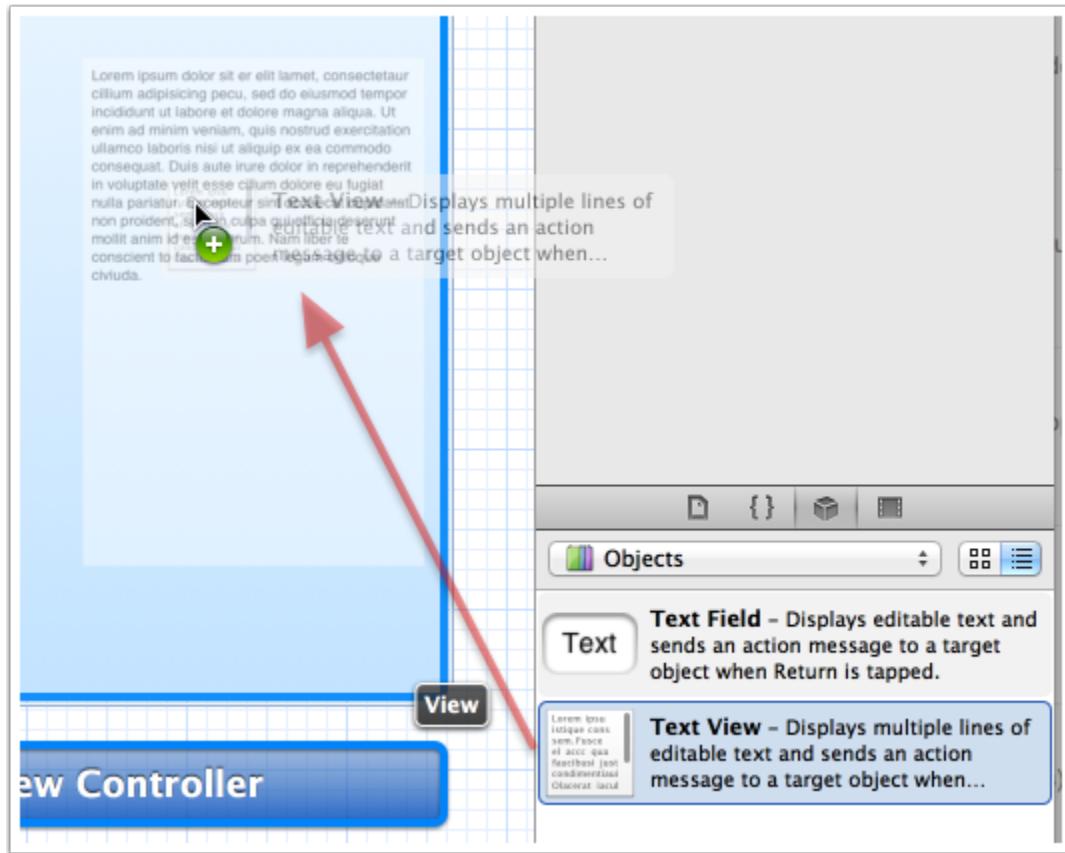
Adding UI Objects to ViewController



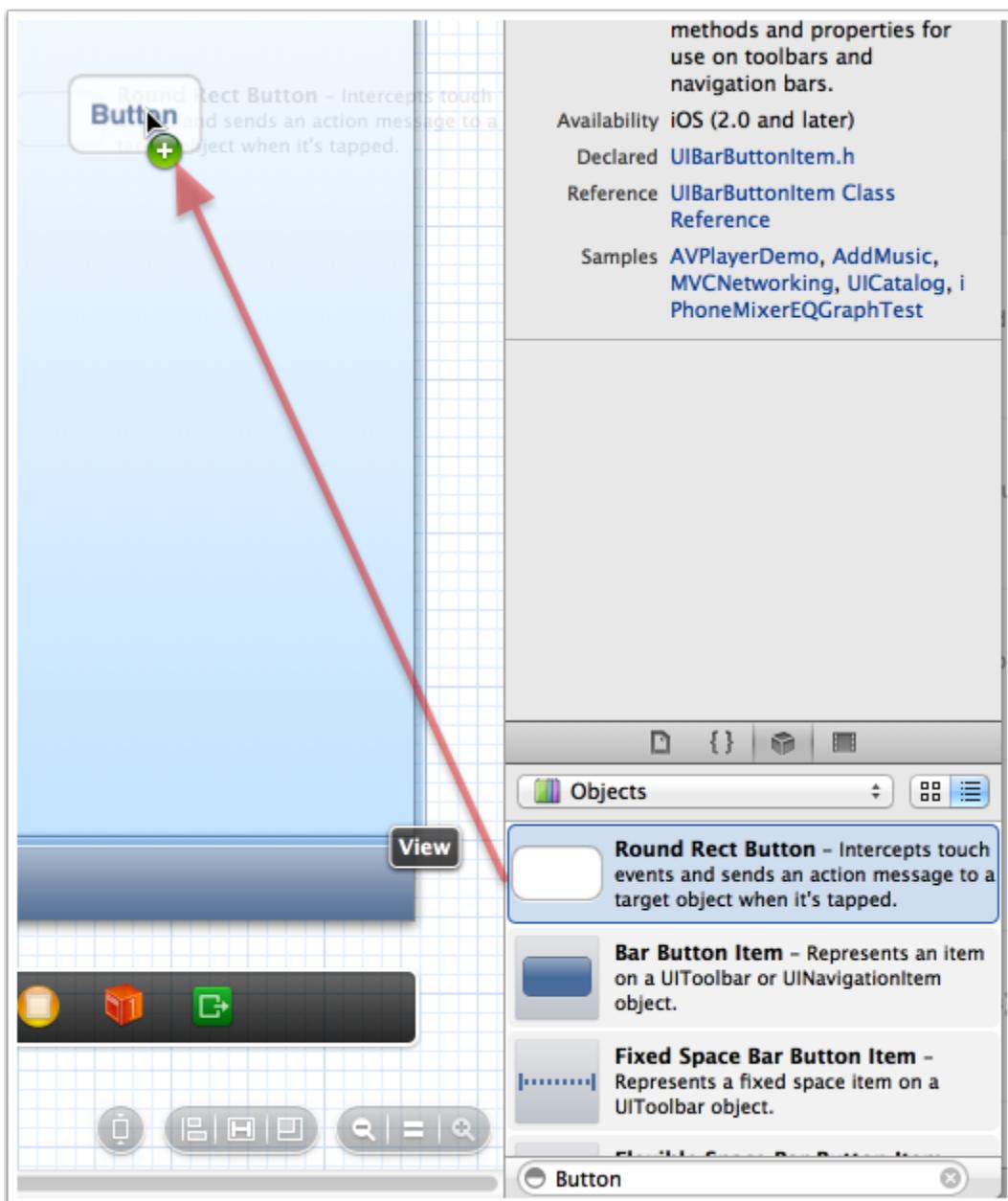
Drag UIKit Object to ViewController - UITextField



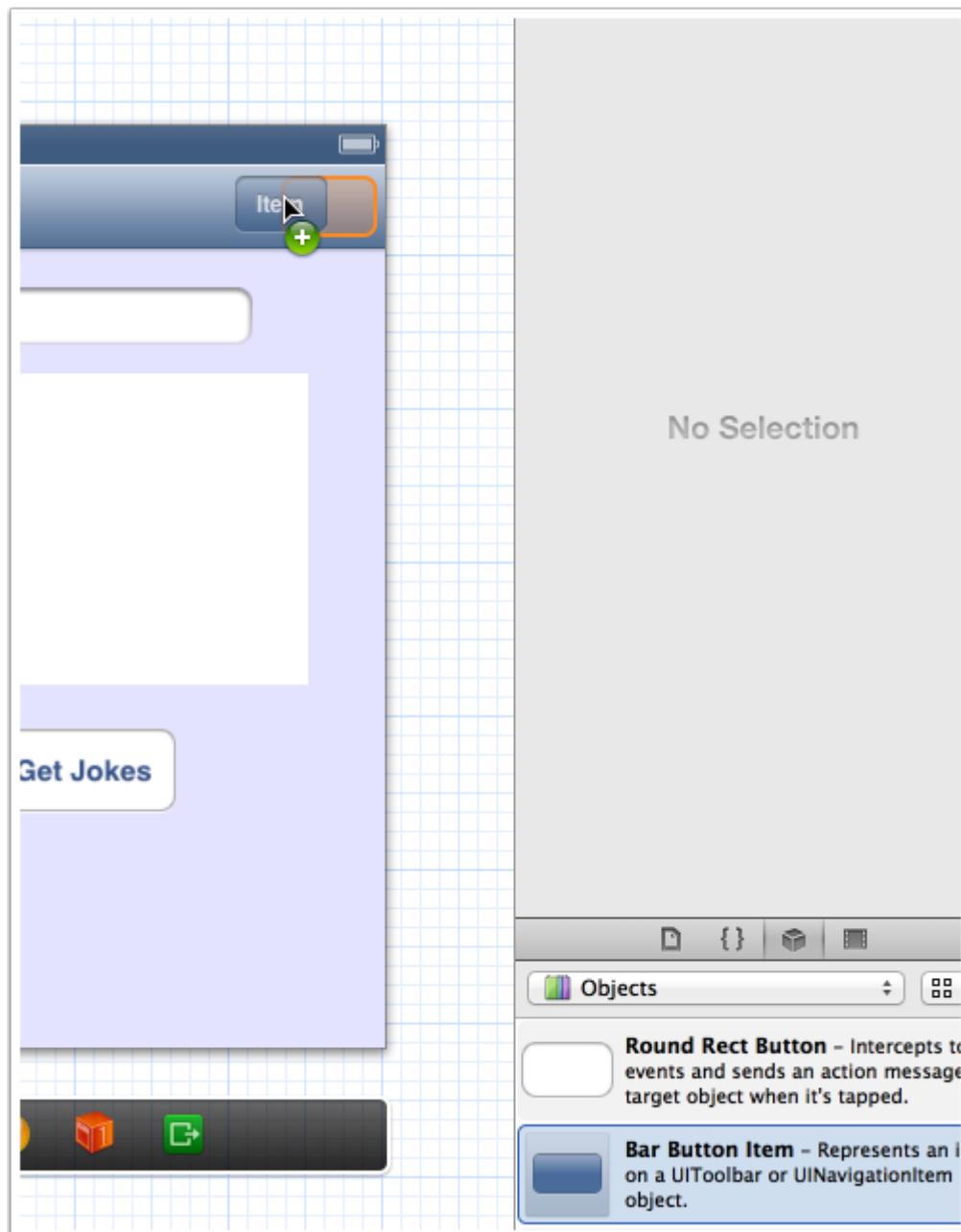
Drag UIKit Object to ViewController (UITextView)



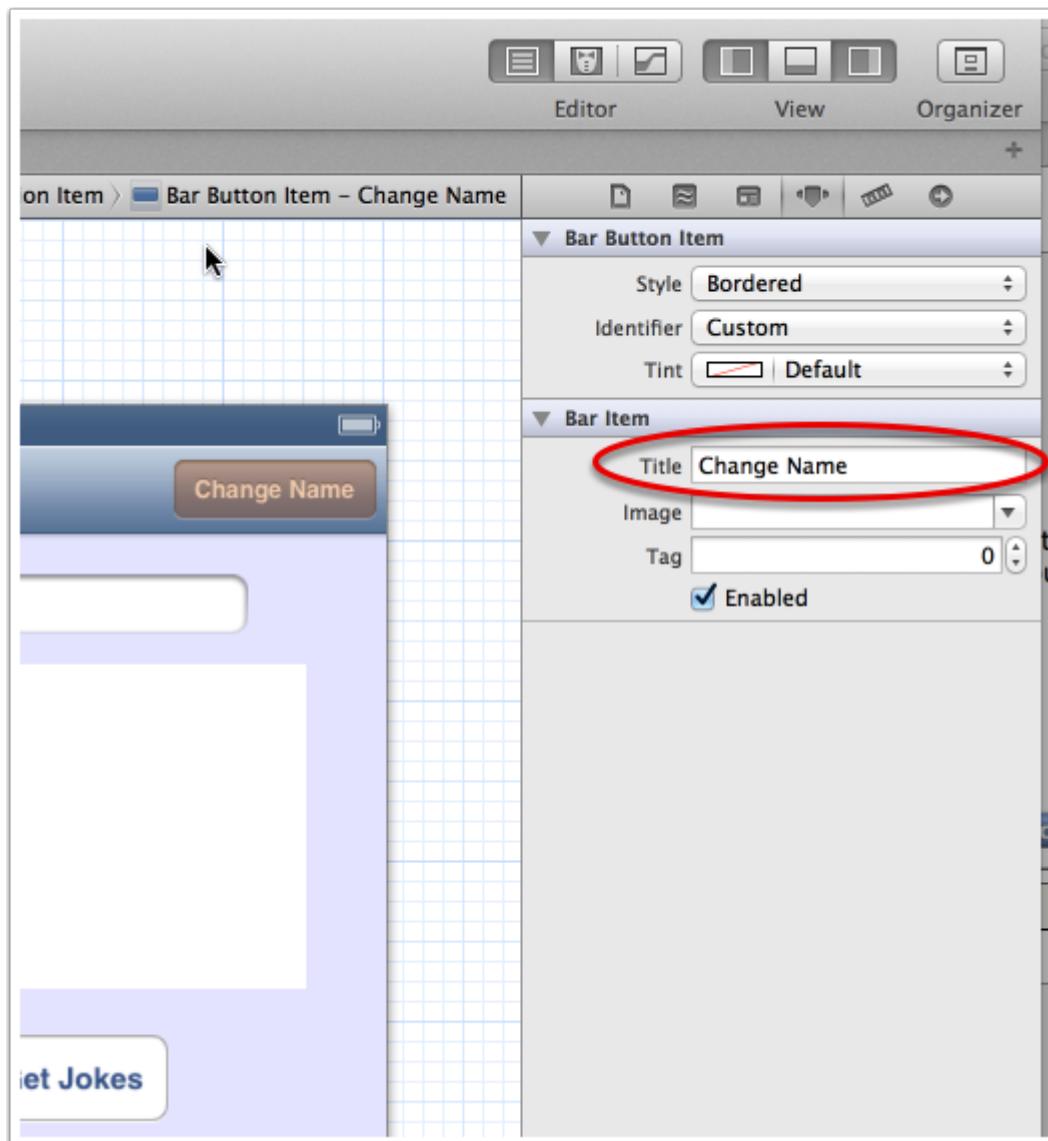
Drag UIKit Object to ViewController (UIButton)



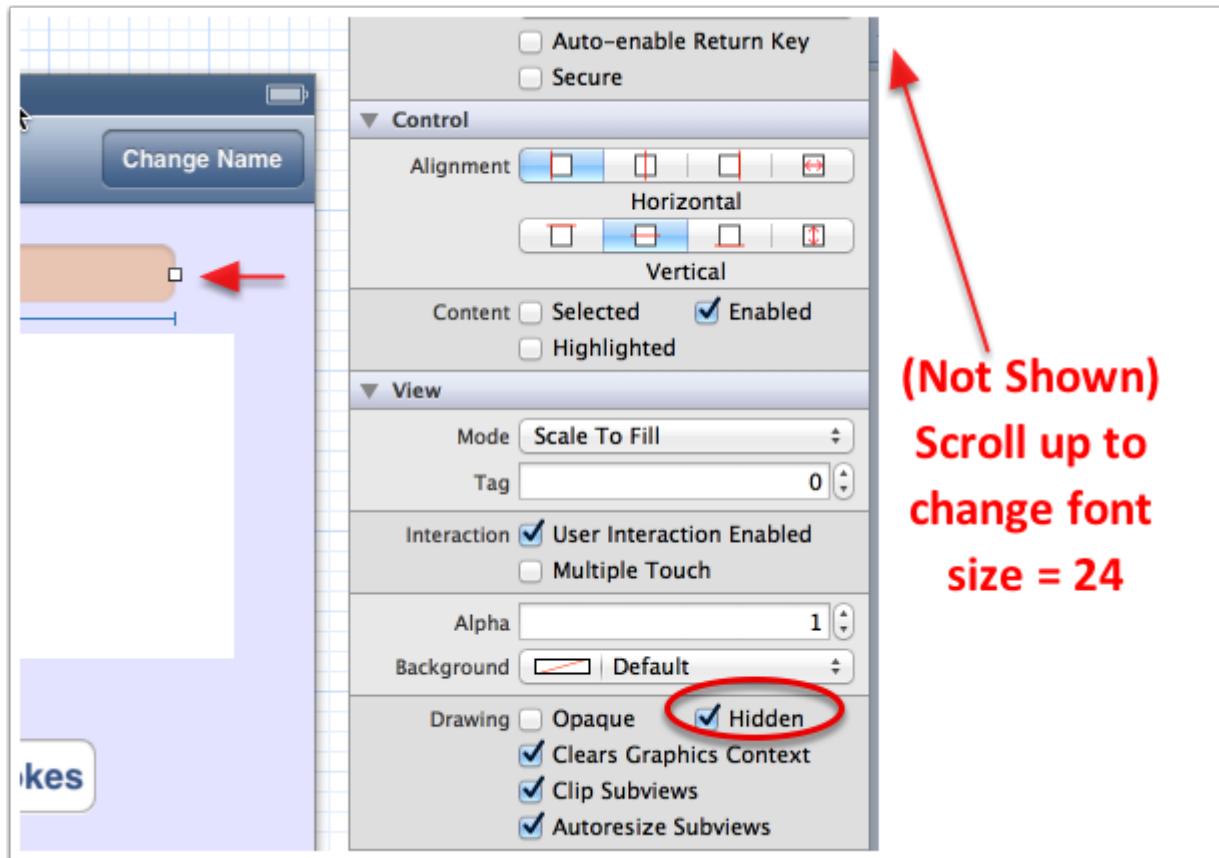
UINav Bar Button



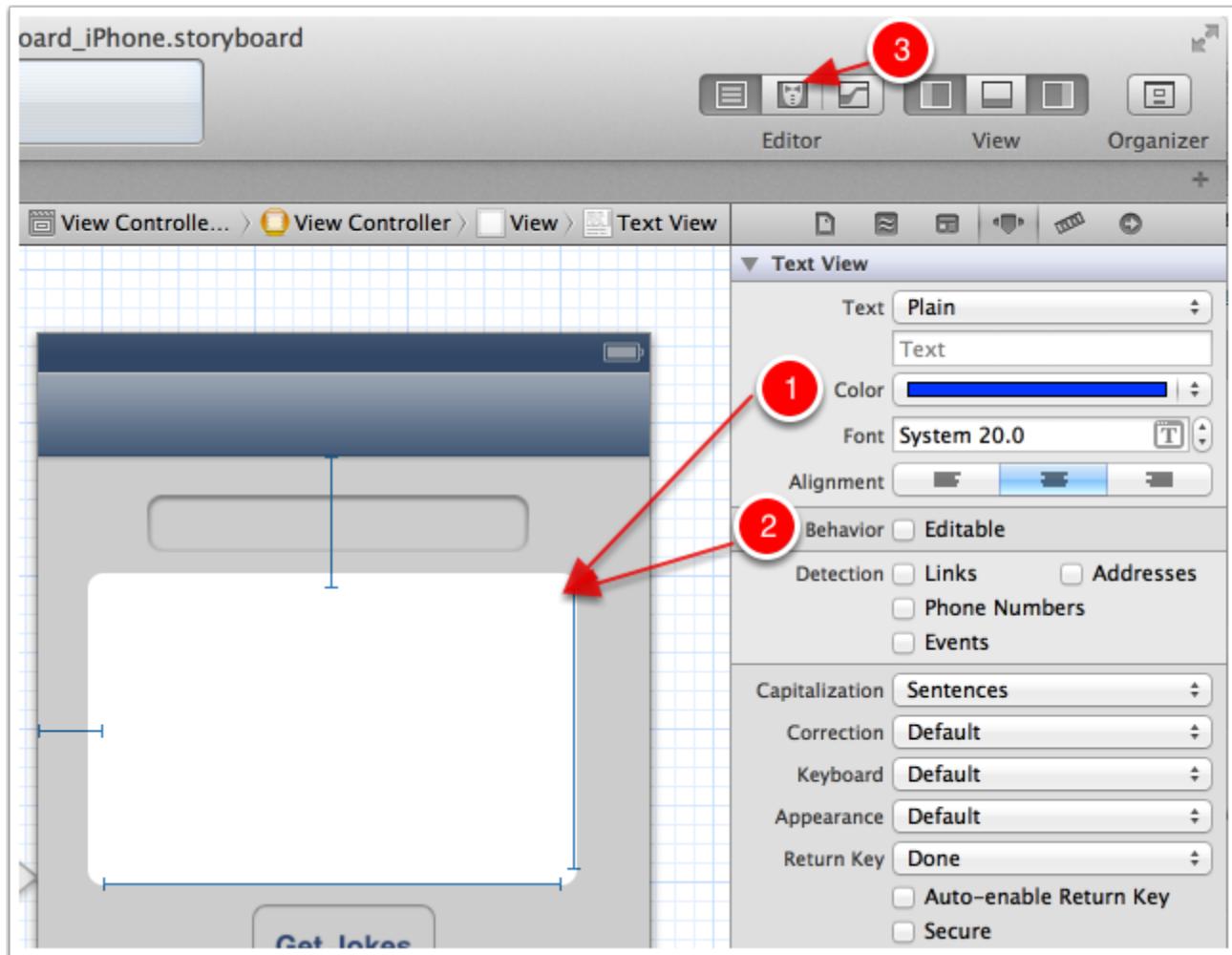
Change Name of Button



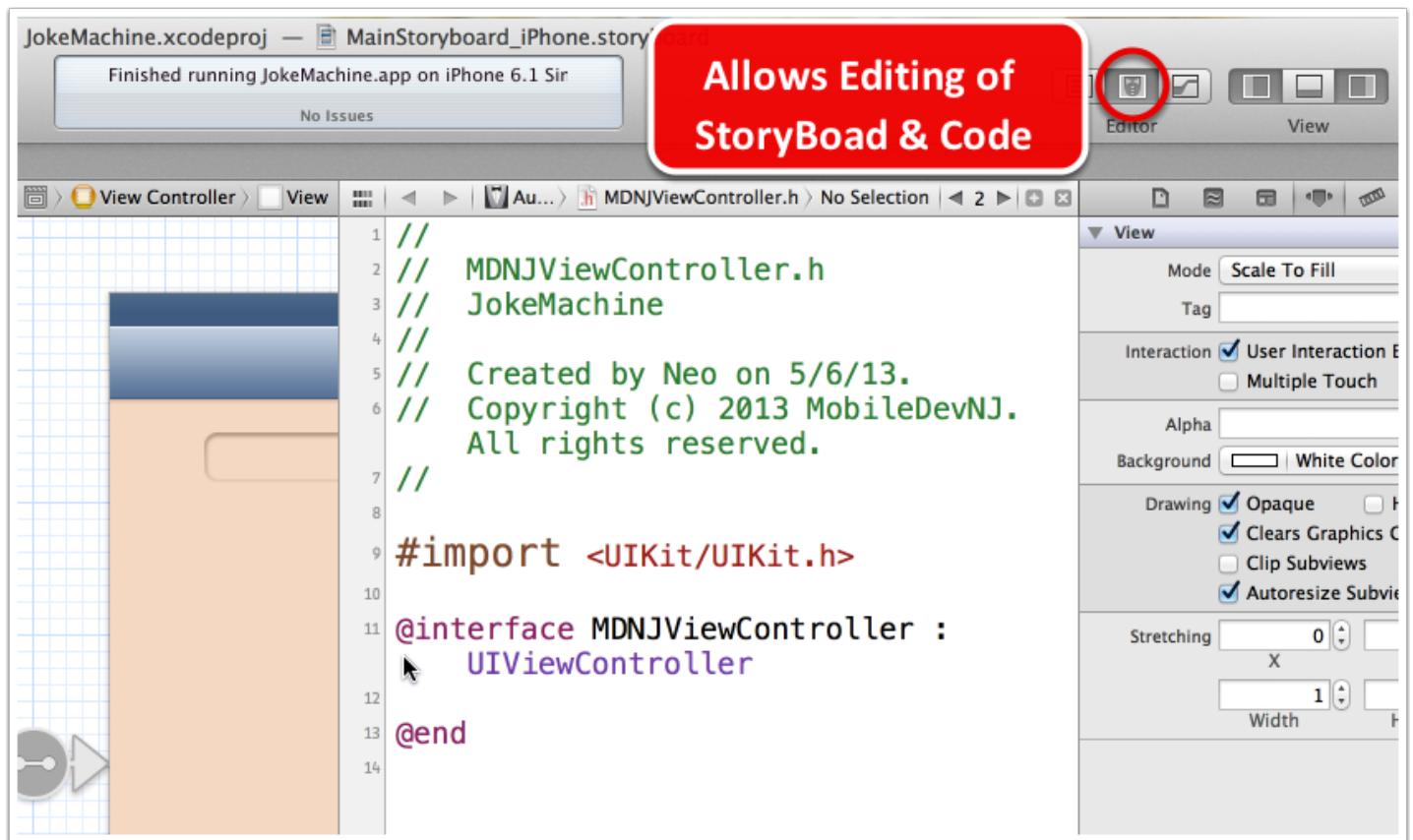
Change TextField to be Hidden by Default



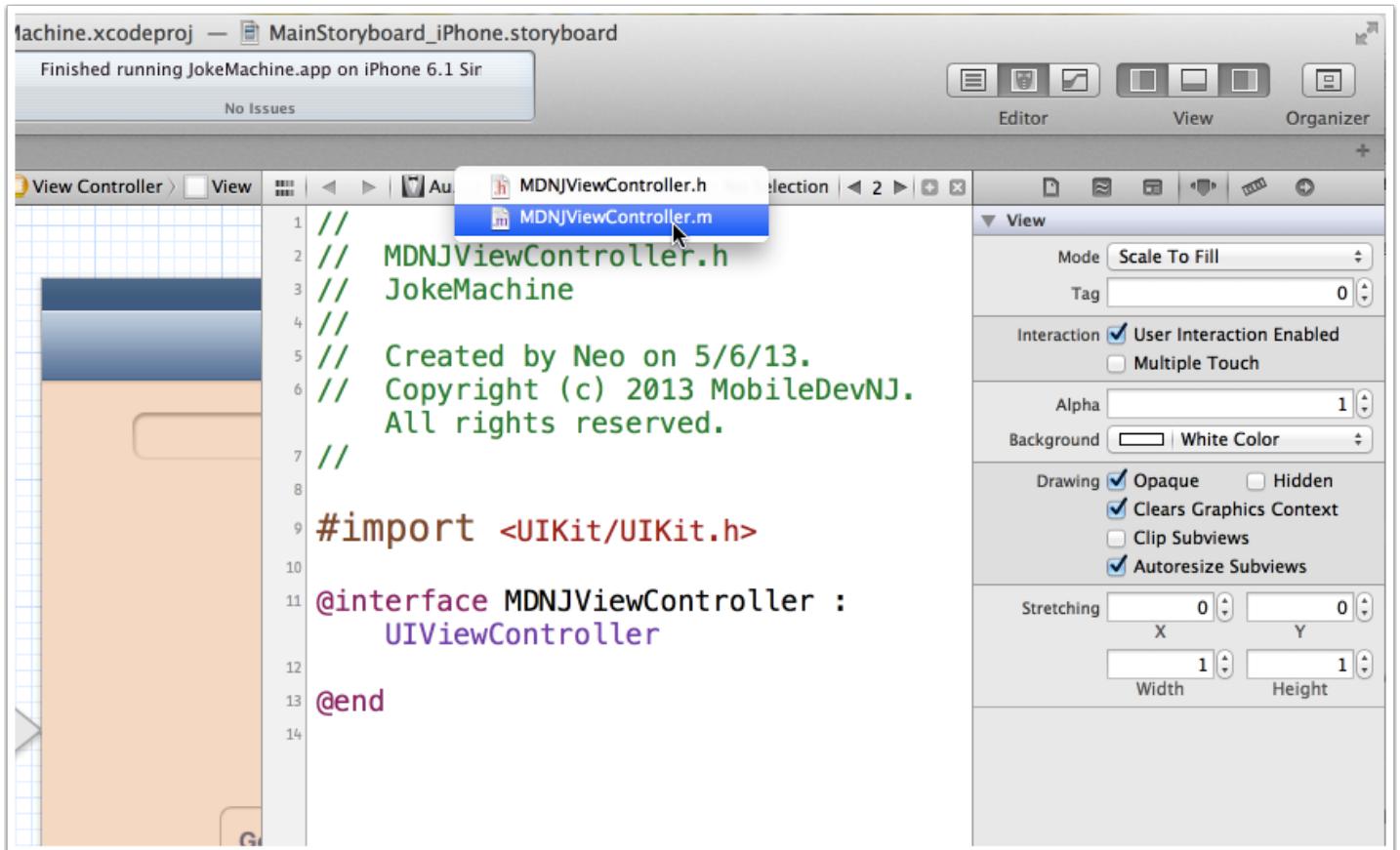
Changing Attributes for UIKit Objects



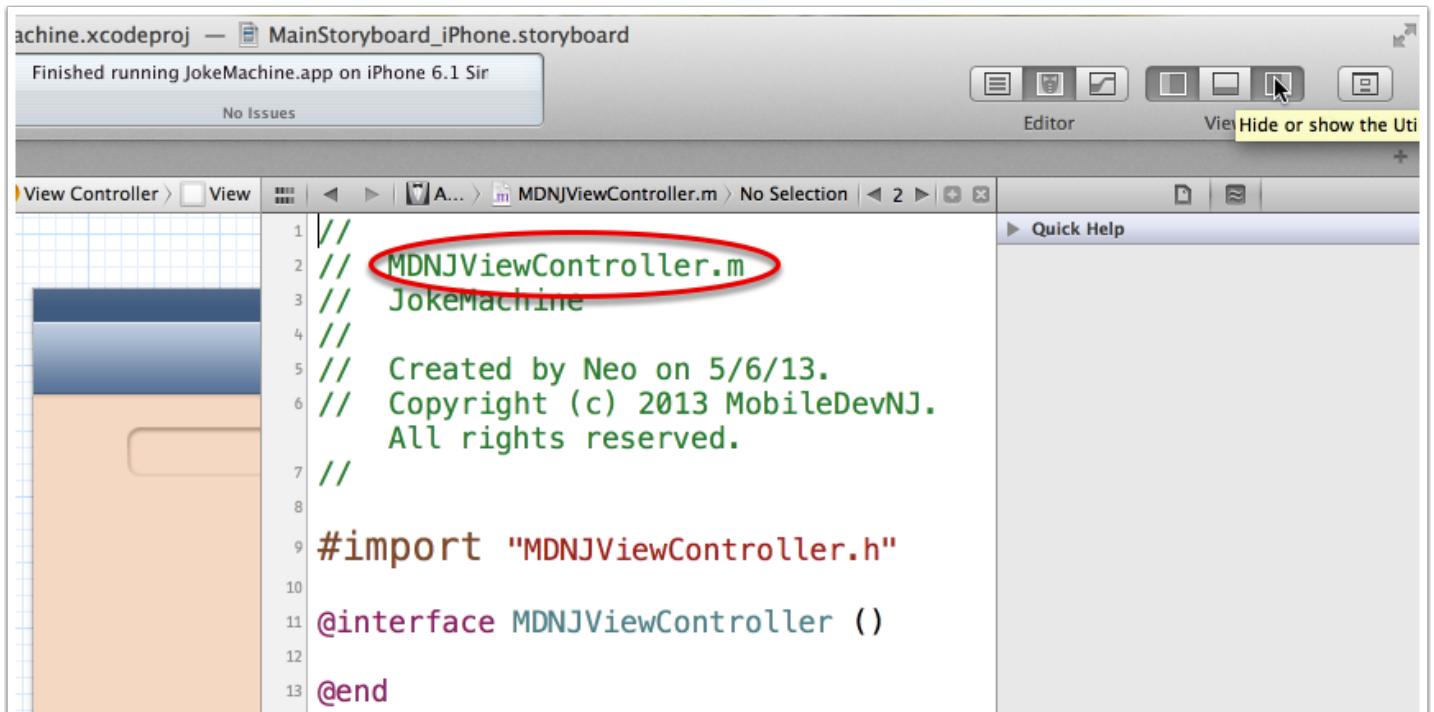
Creating Code from Storyboard



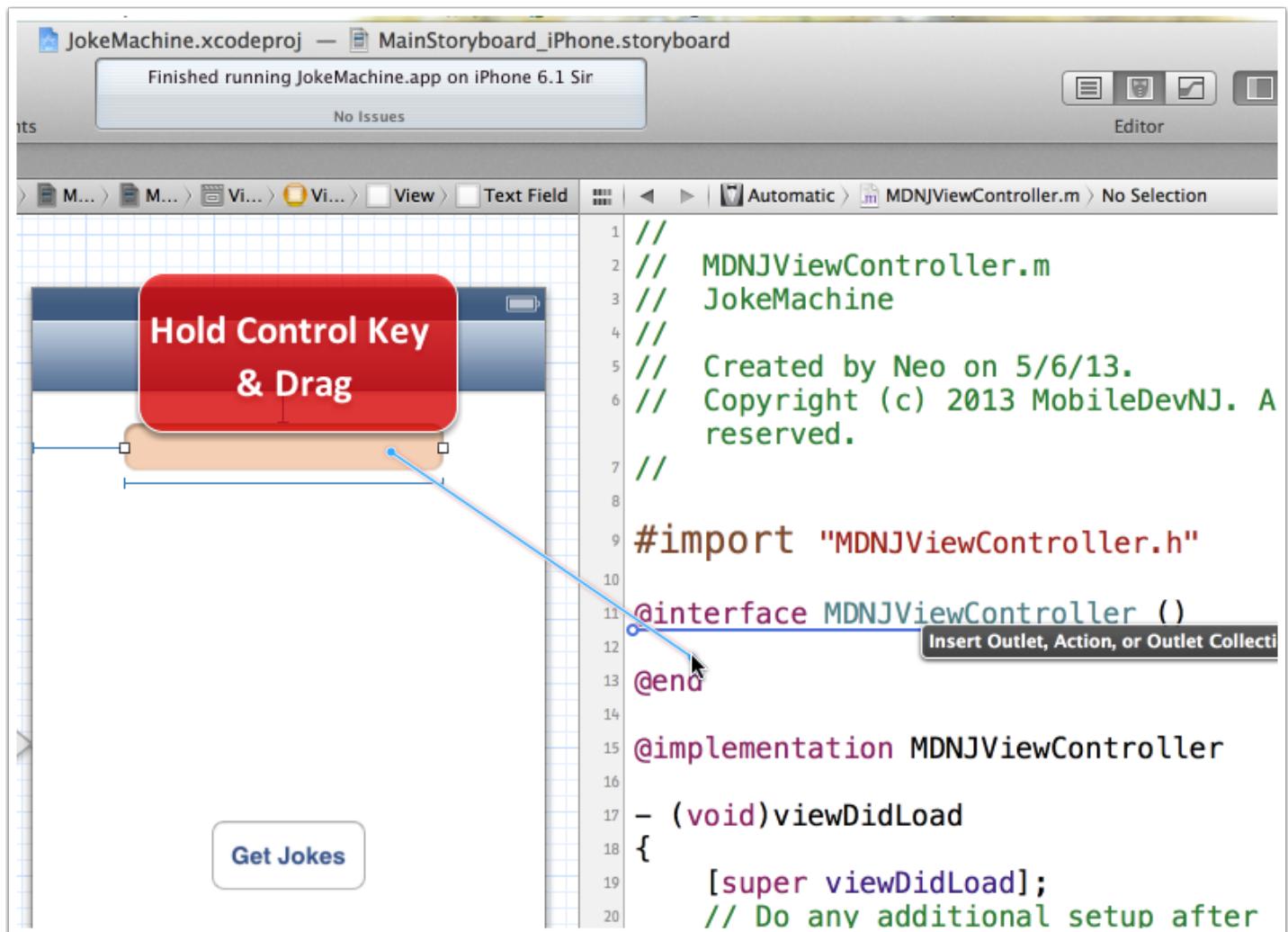
Switching to .m File



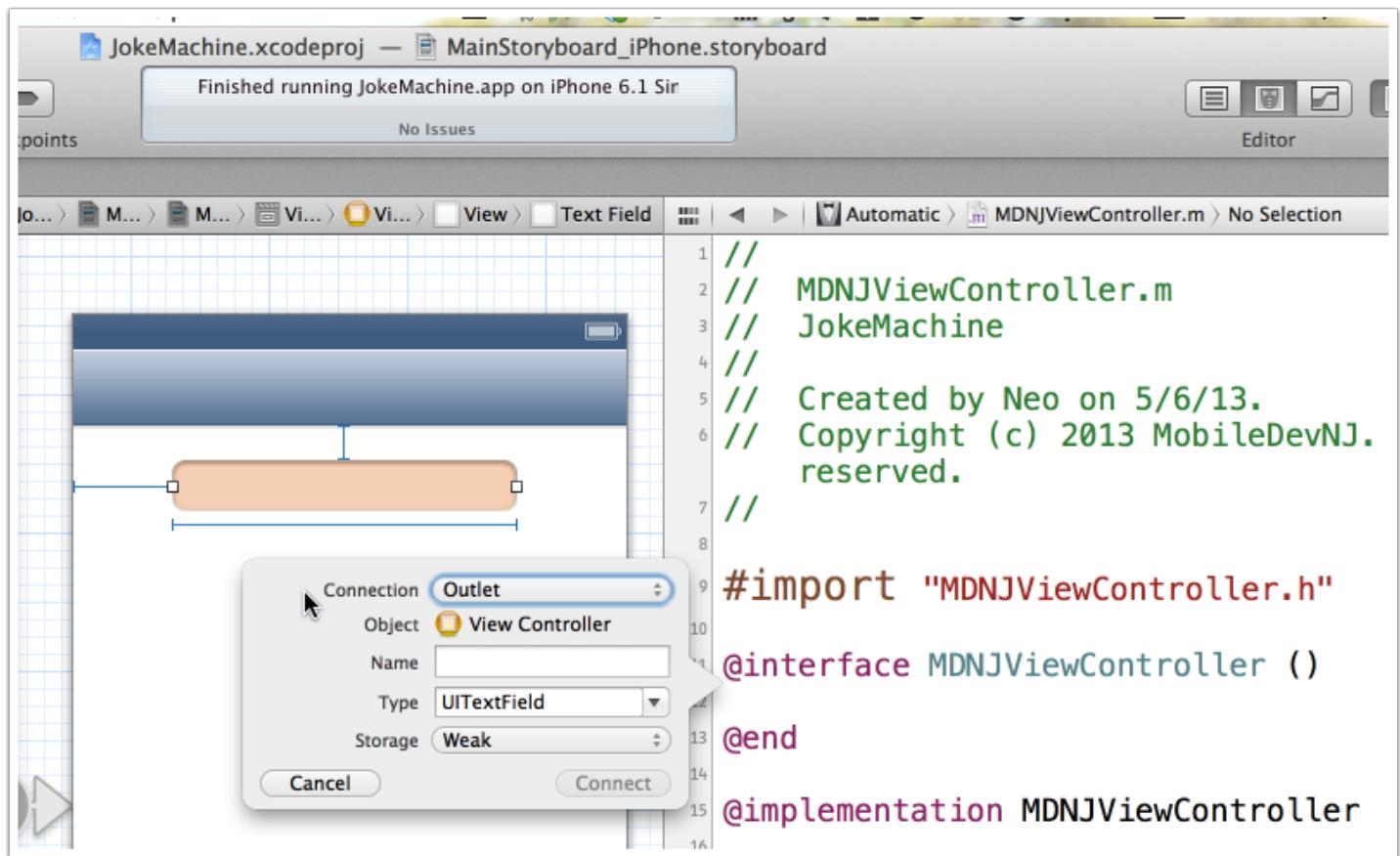
Split Window allows IBOutlet/IBAction Linking



Drag UIKit Object to .m File (outlet or action)



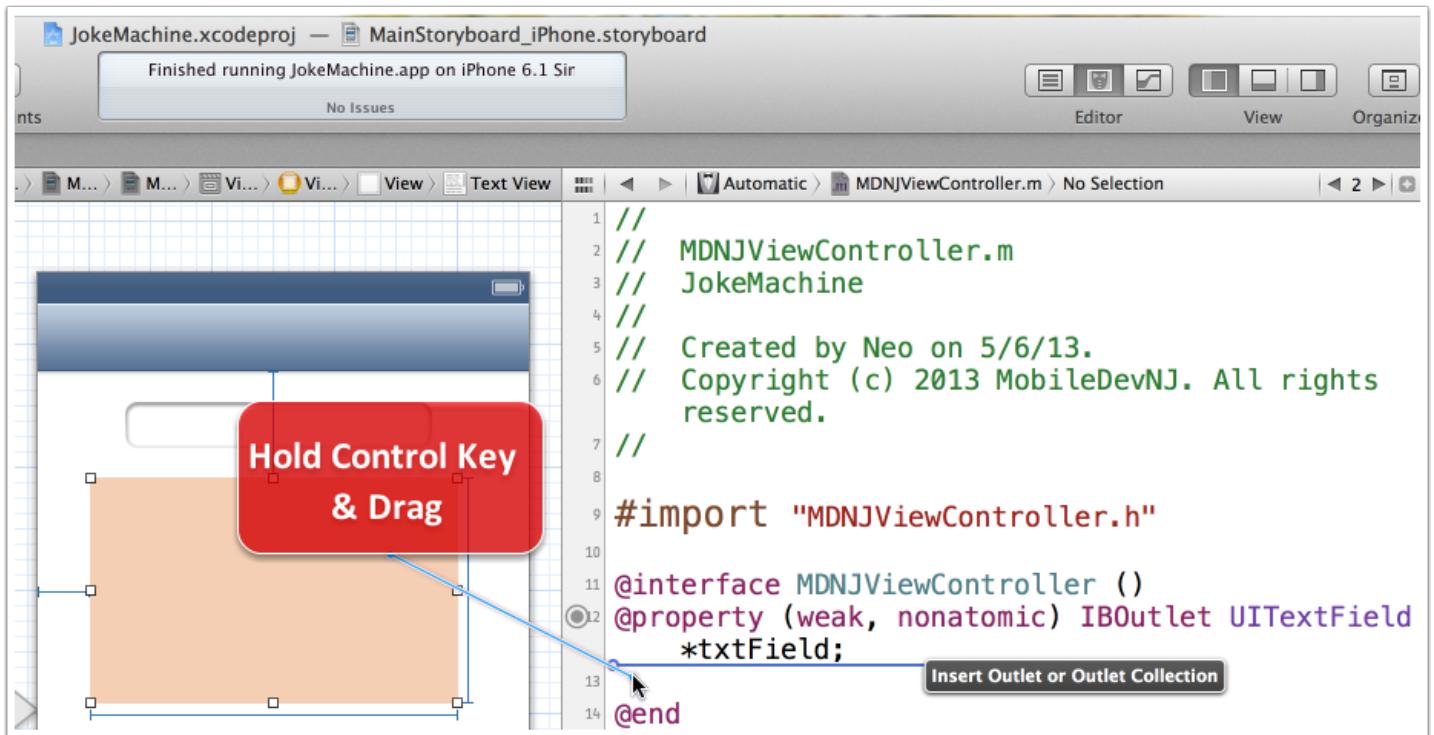
Drag UIKit Object to .m File (naming & type)



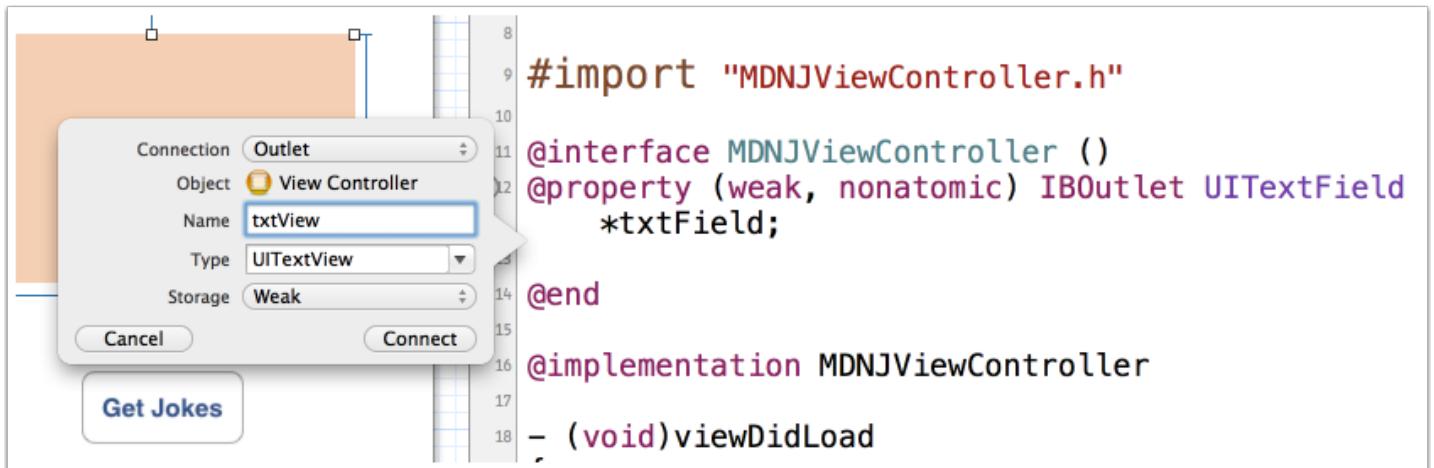
Drag UIKit Object to .m File (property defined)



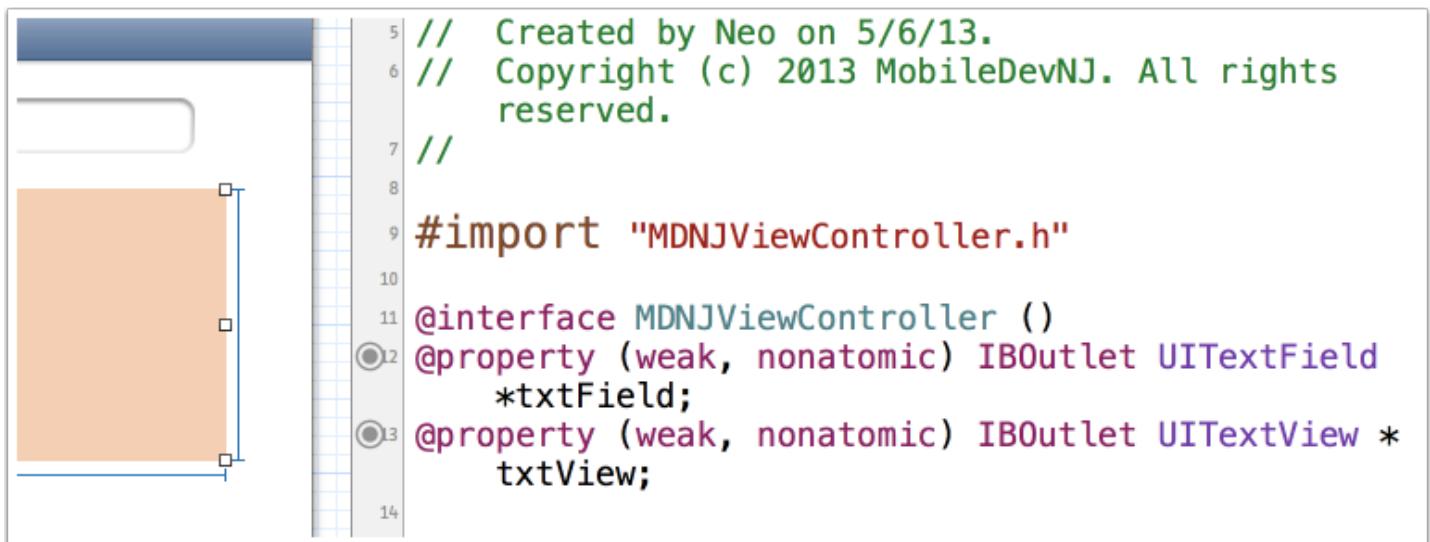
Drag UIKit Object to .m File (defining property)



Drag UIKit Object to .m File (defining property)

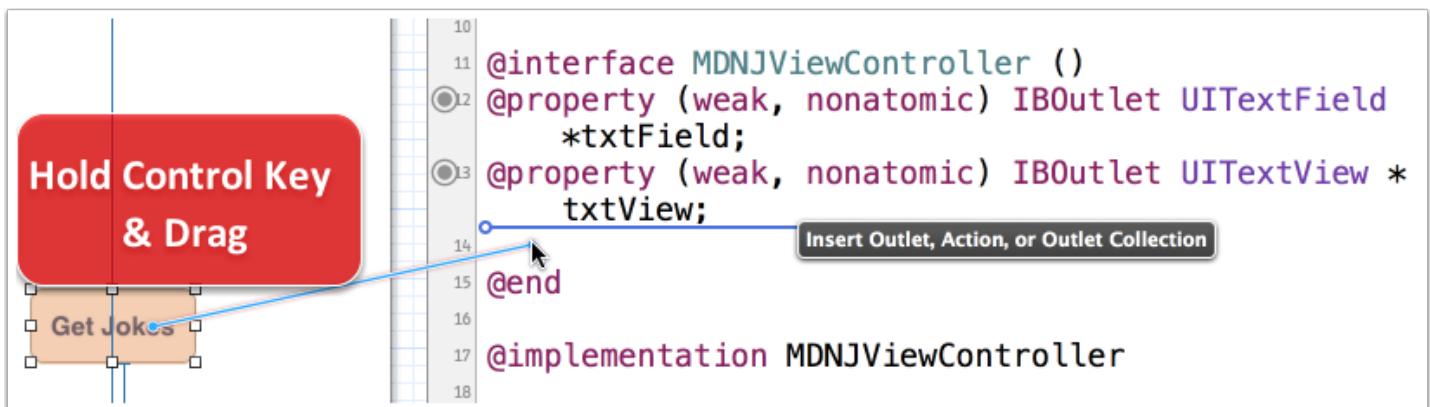


Drag UIKit Object to .m File (defining property)



```
5 // Created by Neo on 5/6/13.  
6 // Copyright (c) 2013 MobileDevNJ. All rights  
7 //  
8  
9 #import "MDNJViewController.h"  
10  
11 @interface MDNJViewController ()  
12 @property (weak, nonatomic) IBOutlet UITextField  
13 *txtField;  
14 @property (weak, nonatomic) IBOutlet UITextView *  
15 txtView;
```

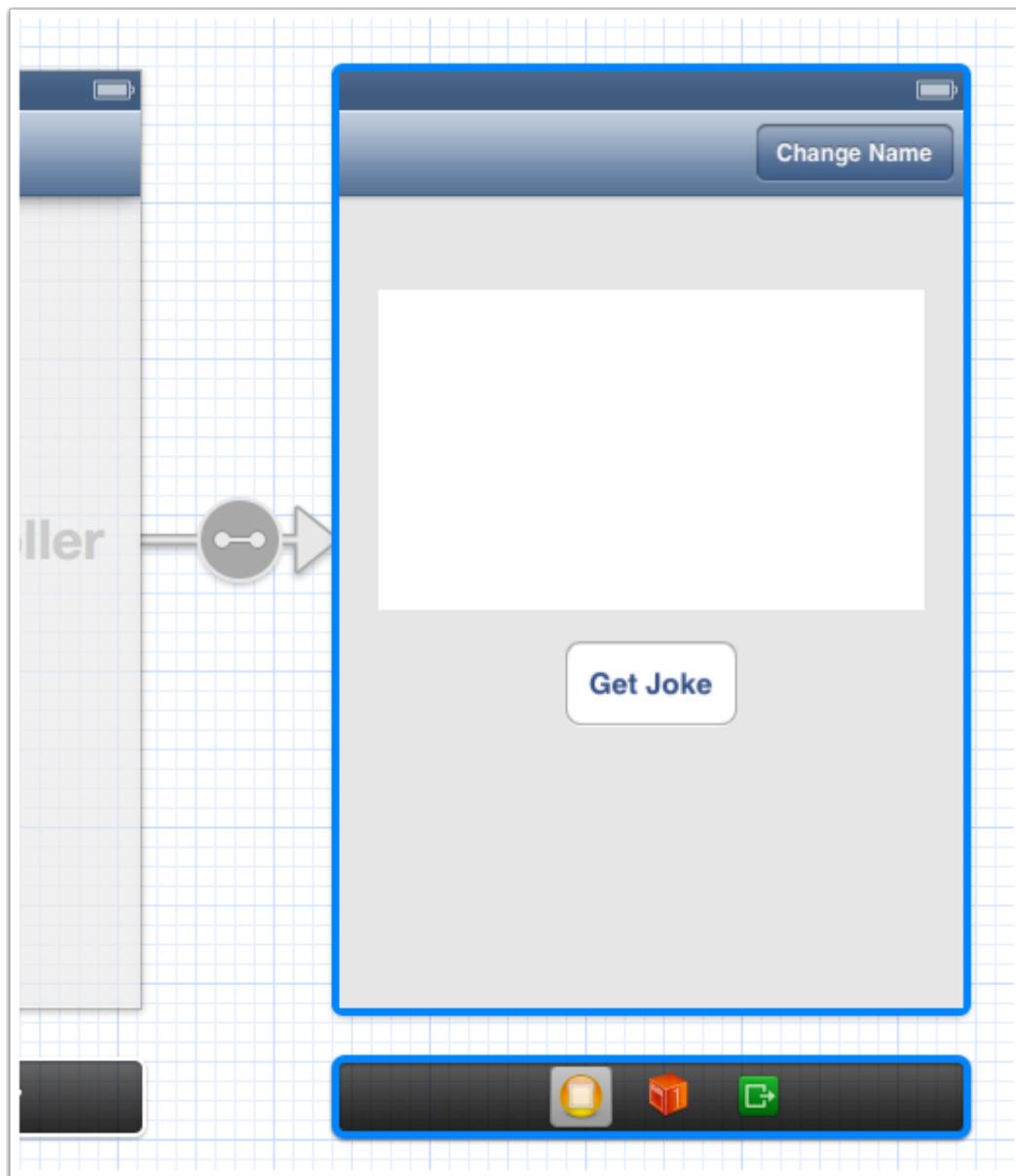
Drag UIKit Object to .m File (defining property)



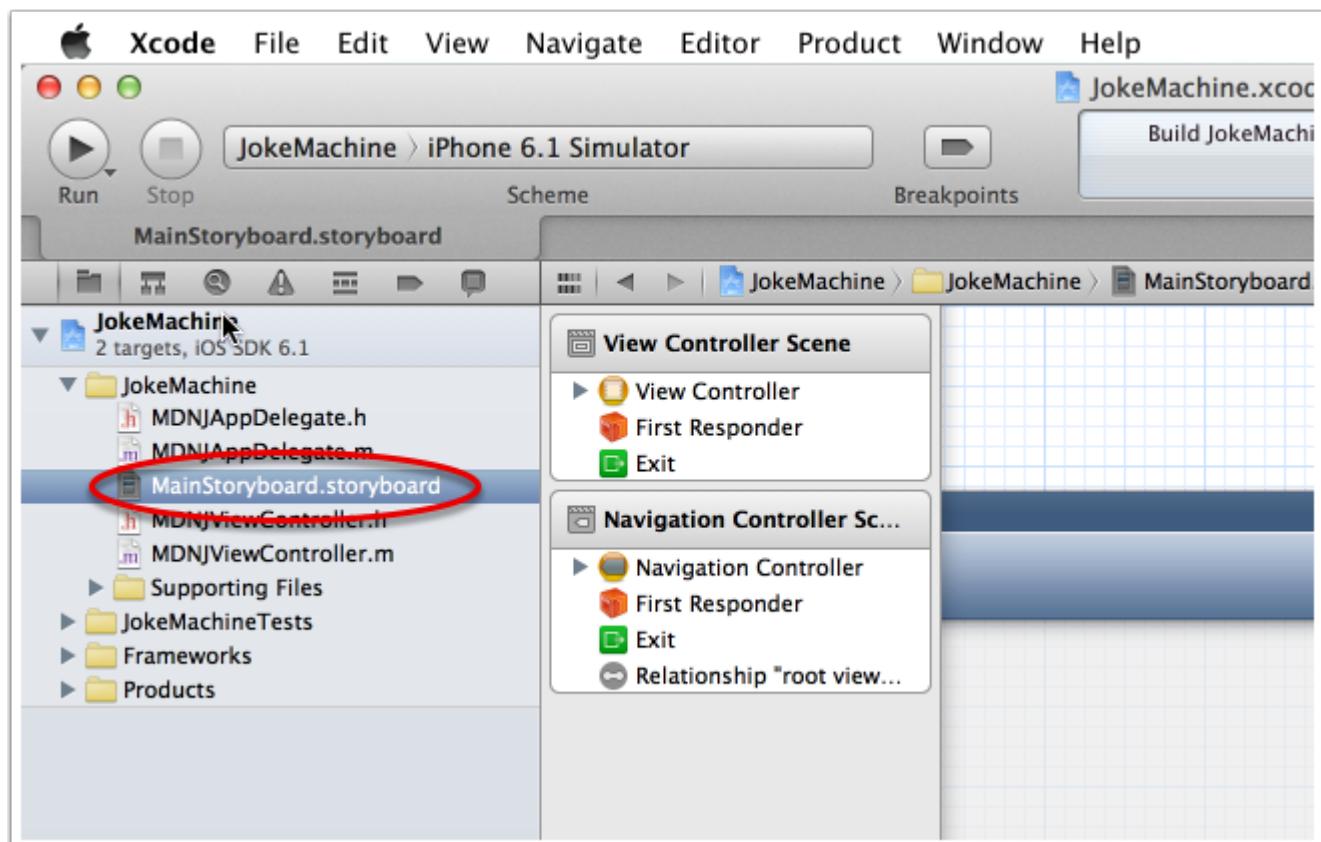
Repeat for btnChangeName

Do the Same for btnChangeName!

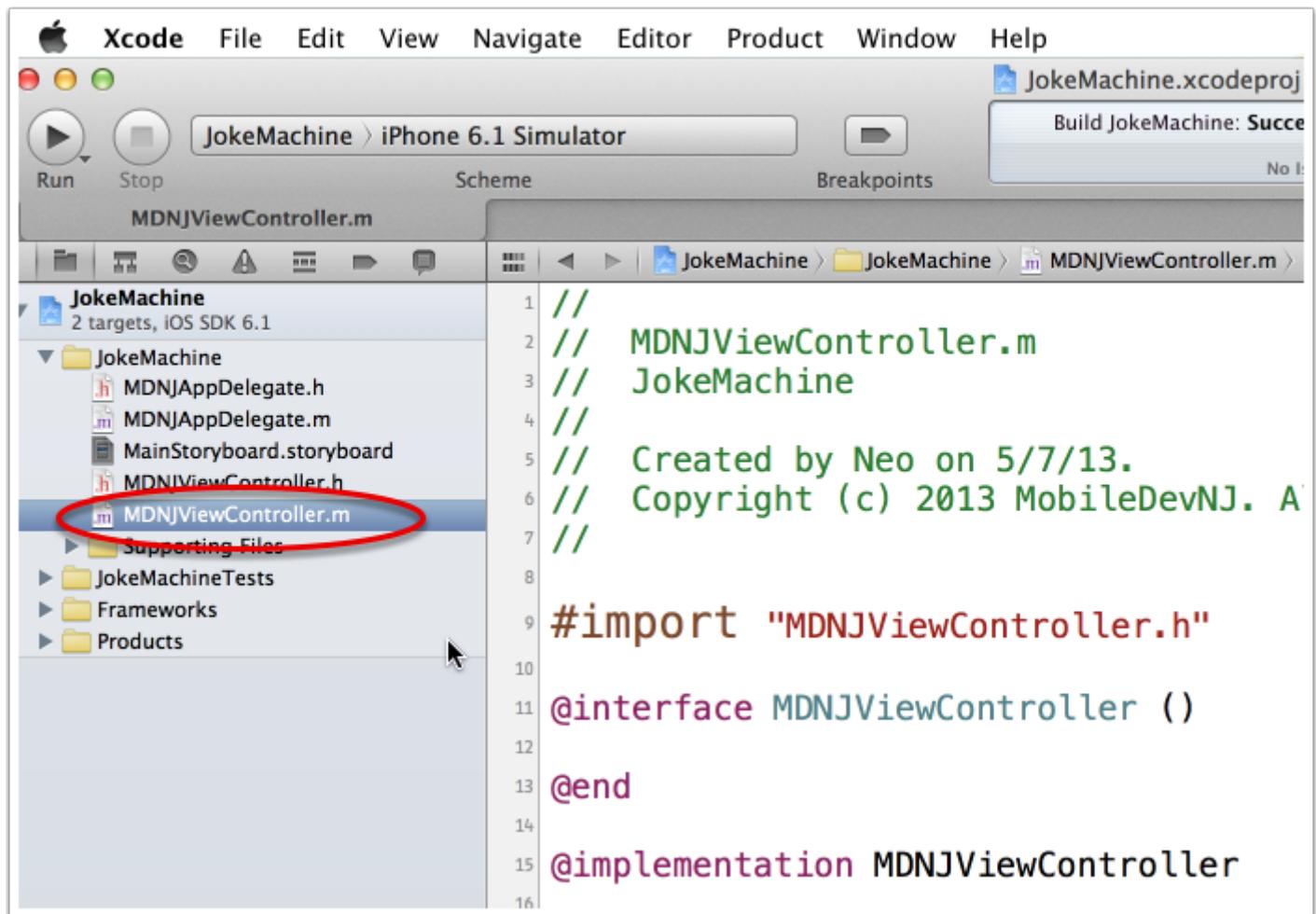
ViewController



Xcode - Switch from Storyboard



Xcode - Switch to Code Editor



The screenshot shows the Xcode interface with the following details:

- Toolbar:** Xcode, File, Edit, View, Navigate, Editor, Product, Window, Help.
- Run/Stop Buttons:** Run, Stop.
- Scheme:** JokeMachine > iPhone 6.1 Simulator.
- Breakpoints:** Breakpoints button.
- Project Navigator:** Shows the project structure for "JokeMachine". The file "MDNJViewController.m" is selected and highlighted with a red oval.
- Code Editor:** Displays the content of "MDNJViewController.m". The code includes comments about the file being created by Neo on 5/7/13 and copyright information for MobileDevNJ. A portion of the code is shown below:

```
// MDNJViewController.m
// JokeMachine
//
// Created by Neo on 5/7/13.
// Copyright (c) 2013 MobileDevNJ. A
//
#import "MDNJViewController.h"

@interface MDNJViewController : UIViewController

@end

@implementation MDNJViewController
```

NSString Object myName

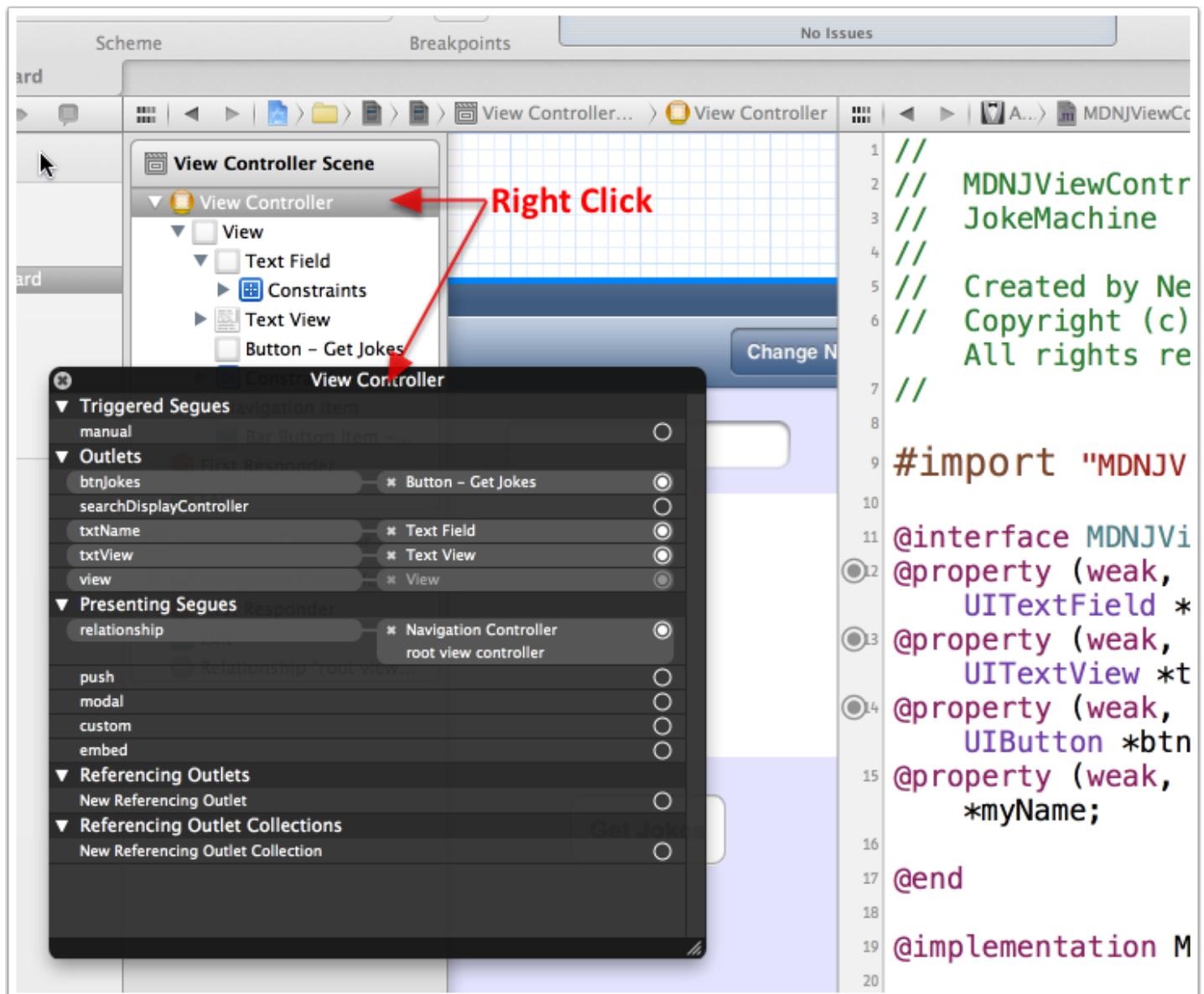
Good time to add Model to App!

**@property (strong, nonatomic) NSString
*myName;**

UI Model Automatically Defined...(although not persisted)

```
5 // Created by Neo on 5/6/13.  
6 // Copyright (c) 2013 MobileDevNJ. All rights reserved.  
7 //  
8  
9 #import "MDNJViewController.h"  
10  
11 @interface MDNJViewController ()  
12 @property (weak, nonatomic) IBOutlet UITextField *txtField;  
13 @property (weak, nonatomic) IBOutlet UITextView *txtView;  
14 @property (weak, nonatomic) IBOutlet UIButton *btnJokes;  
15 @property (strong,nonatomic) NSString *myName;  
16 @end  
17
```

Check Wiring



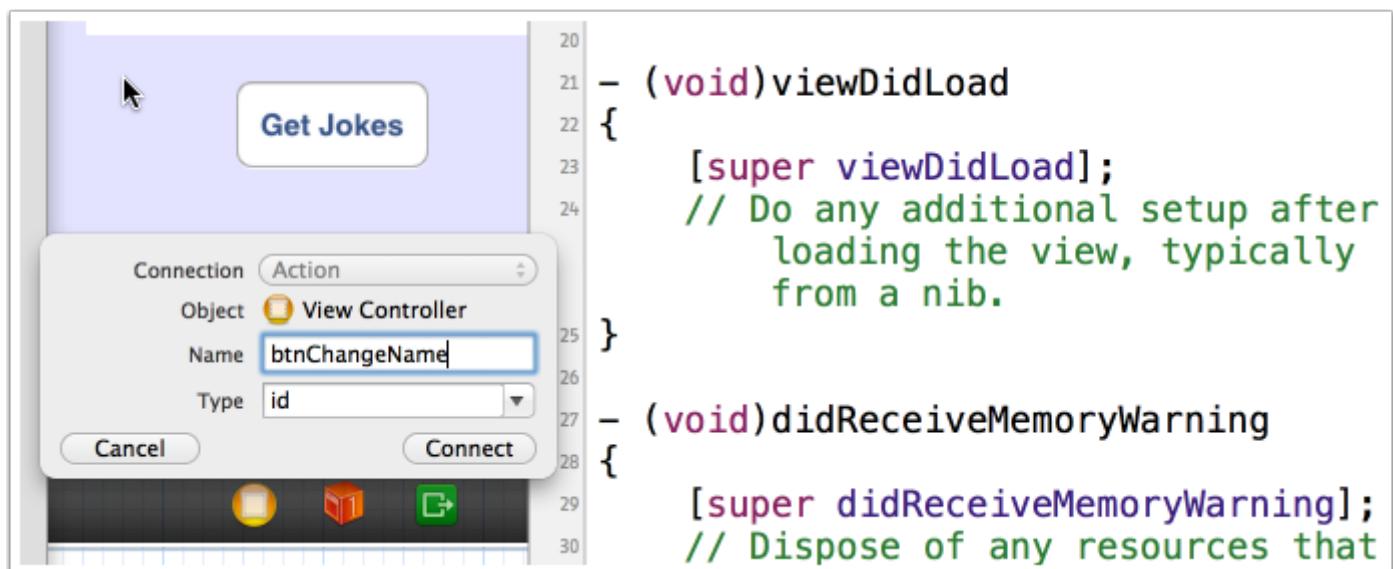
We Demand Action!

We need (3) Actions Defined in View!

1. Name Edit – Remove Keyboard
 - a) didEndOnExit action
2. Get Jokes Button Action
 - a) Touch Up Inside
3. Change Name Button Action
 - a) Touch Up Inside

```
#import "MDNJViewController.h"

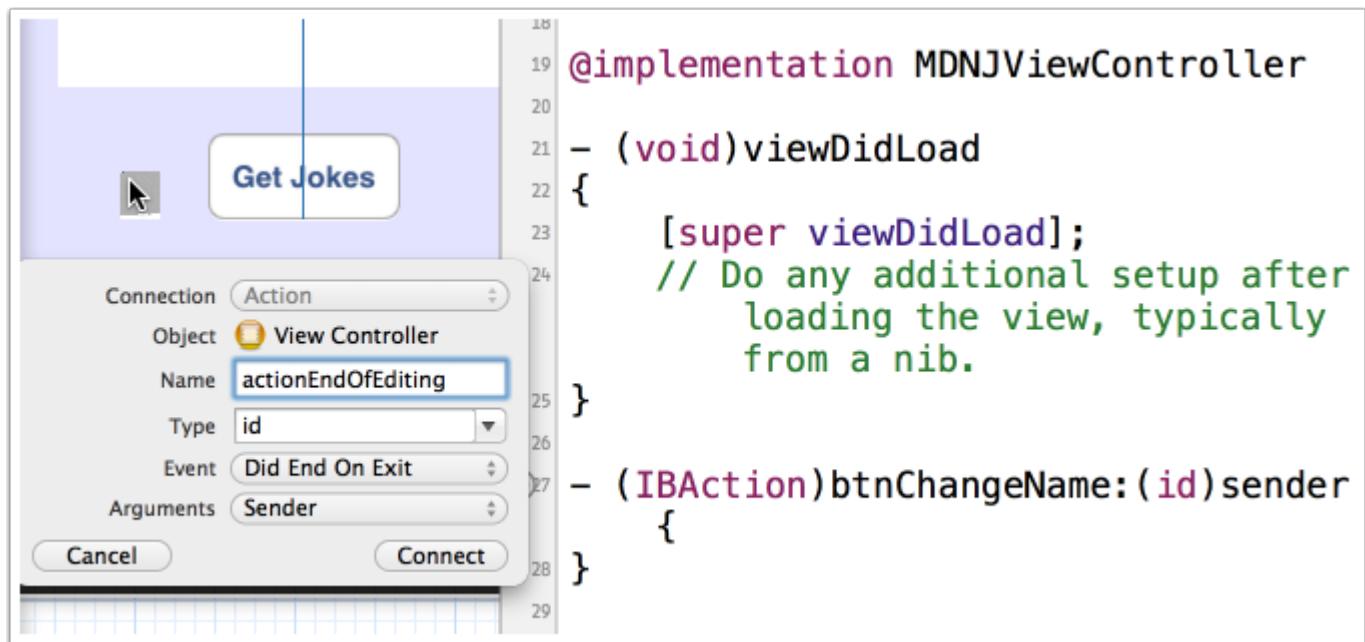
@interface MDNJViewController ()  
@property (weak, nonatomic) IBOutlet  
    UITextField *txtName;  
@property (weak, nonatomic) IBOutlet  
    UITextView *txtView;  
@property (weak, nonatomic) IBOutlet  
    UIButton *btnJokes;  
@property (weak, nonatomic) NSString  
    *myName;  
  
@end  
  
@implementation MDNJViewController  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after  
    // loading the view, typically  
    // from a nib.  
}  
  
- (void)didReceiveMemoryWarning  
{
```



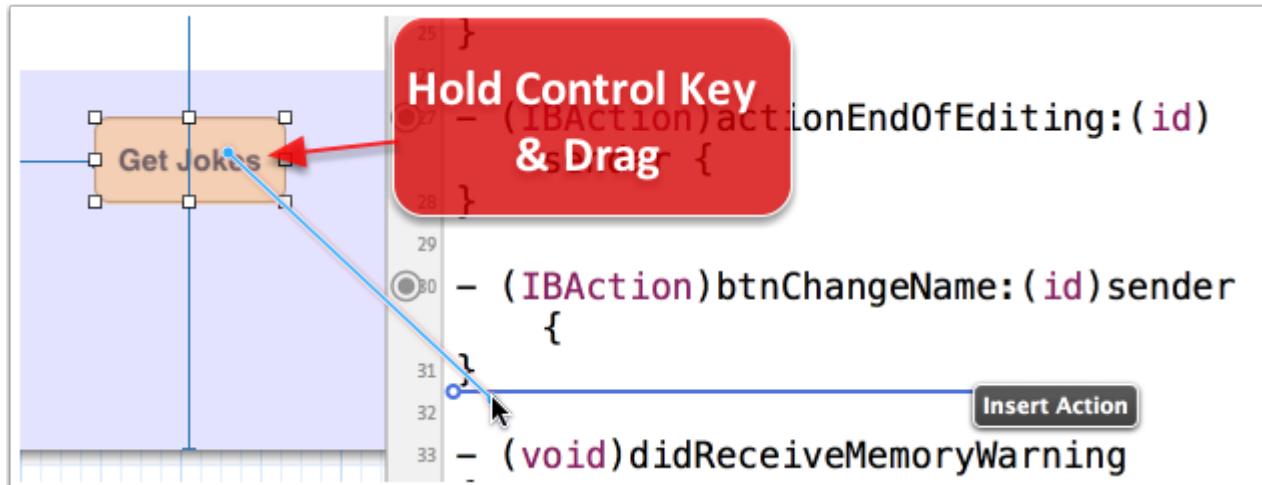
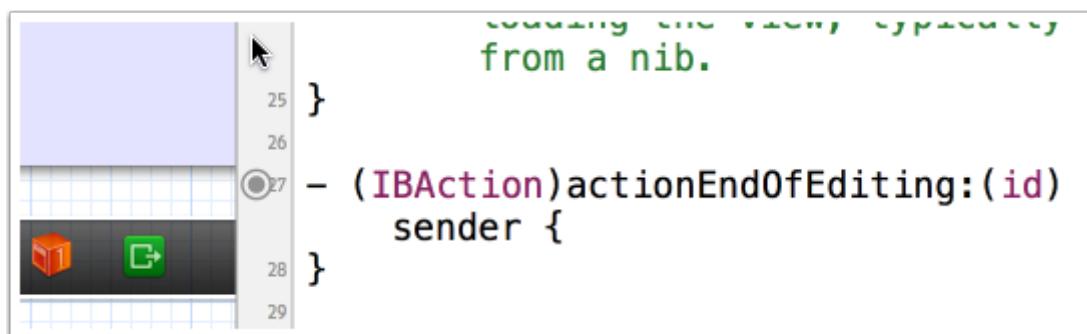
JokeMachine.xcodeproj — MainStoryboard.storyboard

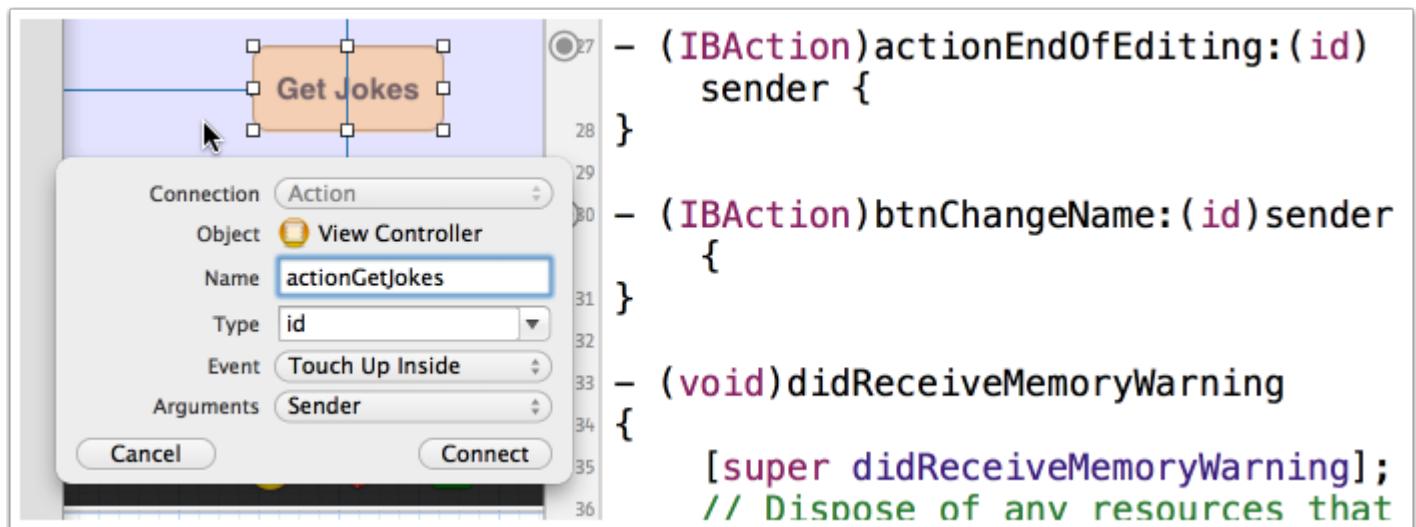
The screenshot shows the Xcode interface with two main windows. The top window is the code editor for `MainStoryboard.storyboard`, displaying Objective-C code for a view controller. The bottom window is the storyboard editor showing a single view with a button labeled "Get Jokes". A red callout bubble with the text "Hold Control Key & Drag" is positioned over the storyboard area, indicating a gesture for connecting outlets or actions. A blue line with arrows shows the connection from the storyboard button to the corresponding line of code in the code editor.

```
// Do any additional setup after loading the view, typically  
// from a nib.  
}  
- (IBAction)btnChangeName:(id)sender  
{  
}  
  
@interface MDNJViewController ()  
@property (weak, nonatomic) IBOutlet UITextField *txtName;  
@property (weak, nonatomic) IBOutlet UITextView *txtView;  
@property (weak, nonatomic) IBOutlet UIButton *btnJokes;  
@property (weak, nonatomic) NSString *myName;  
  
@end  
  
@implementation MDNJViewController  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typical  
    // from a nib.  
}  
- (IBAction)btnChangeName:(id)sender
```



JokeMachine.xcodeproj — MainStoryboard.storyboard





Functions Pre-defined

```

18 @implementation MDNJViewController
19
20 - (void)viewDidLoad
21 {
22     [super viewDidLoad];
23     // Do any additional setup after loading the view, typically from a nib.
24 }
25
26 - (IBAction)btnChangeName:(id)sender {
27 }
28
29
30 - (IBAction)actionEndOfEditing:(UITextField*)sender {
31 }
32
33
34 - (IBAction)actionGetJokes:(id)sender {
35     // Get joke and set text View
36 }
37

```

We Need Property for our Model -> myName

Don't forget to add our "Model" myName!

```
Board 8  
9 #import "MDNJViewController.h"  
10  
11 @interface MDNJViewController()  
12 @property (weak, nonatomic) IBOutlet UITextField *txtName;  
13 @property (weak, nonatomic) IBOutlet UITextView *txtView;  
14 @property (weak, nonatomic) IBOutlet UIButton *btnJokes;  
15  
16 @property (strong, nonatomic) NSString *myName;  
17  
18 @end  
19  
20 @implementation MDNJViewController  
21 - (void)viewDidLoad  
22 {  
23     [super viewDidLoad];  
24     // Do any additional setup after loading the view, typically from a nib.  
25     self.myName = [[NSString alloc] init];  
26 }  
27  
28
```

Initialize myName!
with Alloc/Init

We Initialize Objects before we use them!

NSString *firstName = [[NSString alloc] init];

NSMutableArray *myArray =

[[NSMutableArray alloc] init];

NSDictionary *myDict = [[NSDictionary alloc] init];

Not Primitives though!

float pi = 3.14;

int age = 25;

JokeMachine.xcodeproj — MDNJViewController.m

```
9 #import "MDNJViewController.h"
10
11 @interface MDNJViewController ()
12 @property (weak, nonatomic) IBOutlet UITextField *txtName;
13 @property (weak, nonatomic) IBOutlet UITextView *txtView;
14 @property (weak, nonatomic) IBOutlet UIButton *btnJokes;
15
16 @property (strong, nonatomic) NSString *myName;
17
18 @end
19
20 @implementation MDNJViewController
21
22 - (void)viewDidLoad
23 {
24     [super viewDidLoad];
25     // Do any additional setup after loading the view, typically from a nib.
26     self.myName = [[NSString alloc] init];
27 }
28
```

Functions Pre-defined

```
18 @implementation MDNJViewController
19
20 - (void)viewDidLoad
21 {
22     [super viewDidLoad];
23     // Do any additional setup after loading the view, typically from a nib.
24 }
25
26 - (IBAction)btnChangeName:(id)sender {
27 }
28
29 - (IBAction)actionEndOfEditing:(UITextField*)sender {
30 }
31
32 - (IBAction)actionGetJokes:(id)sender {
33     // Get joke and set text View
34 }
35
36
37 }
```

Method btnChangeName -

```
25  
26 - (IBAction)btnChangeName:(id)sender {  
27  
28     self.txtName.hidden=FALSE;  
29     [self.txtName becomeFirstResponder];  
30 }  
31
```

Method actionEndOfEditing -

```
31 - (IBAction)actionEndofEditing:(UITextField*)sender {  
32     self.myName = sender.text;  
33     self.navigationItem.title=self.myName;  
34  
35     [sender resignFirstResponder];  
36     sender.hidden=TRUE;  
37     sender.text=@"";  
38 }
```

Functionality of App

We Need a Bunch of iOS Classes!

1. **NSURL** <- URL for Joke Server API
2. **NSURLRequest** <- HTTPRequest
3. **NSURLConnection** <- Connection
4. **NSData** <- returned data
5. **NSJSONSerialization** <-binary->JSON
6. **NSDictionary** <- JSON unpacked

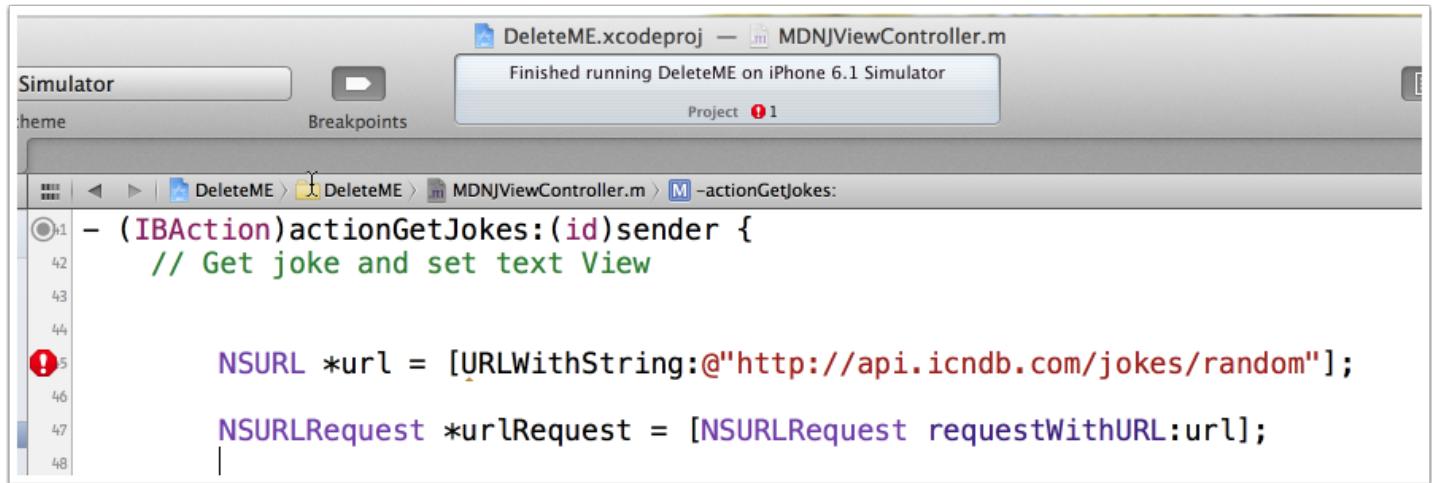
NSURL IDE Completion



A screenshot of an Xcode code editor showing a partial implementation of a class. The code includes several IBAction methods and a didReceiveMemoryWarning method. A cursor is positioned at the start of the word 'NSURL' in the middle of a line, with a blue completion dropdown menu open above it. The menu lists several suggestions starting with 'NSURL': 'NSURL', 'NSURL +', 'NSURL -', 'NSURLConnection', 'NSURLRequest', and 'NSURLResponse'. The line numbers on the left margin range from 28 to 40.

```
28 }
29
30 - (IBAction)btnChangeName:(id)sender {
31 }
32 - (IBAction)actionGetJokes:(id)sender {
33
34     NSURL[Completion Suggestion]
35
36 }
37
38 - (void)didReceiveMemoryWarning
39 }
```

IDE warnings and errors

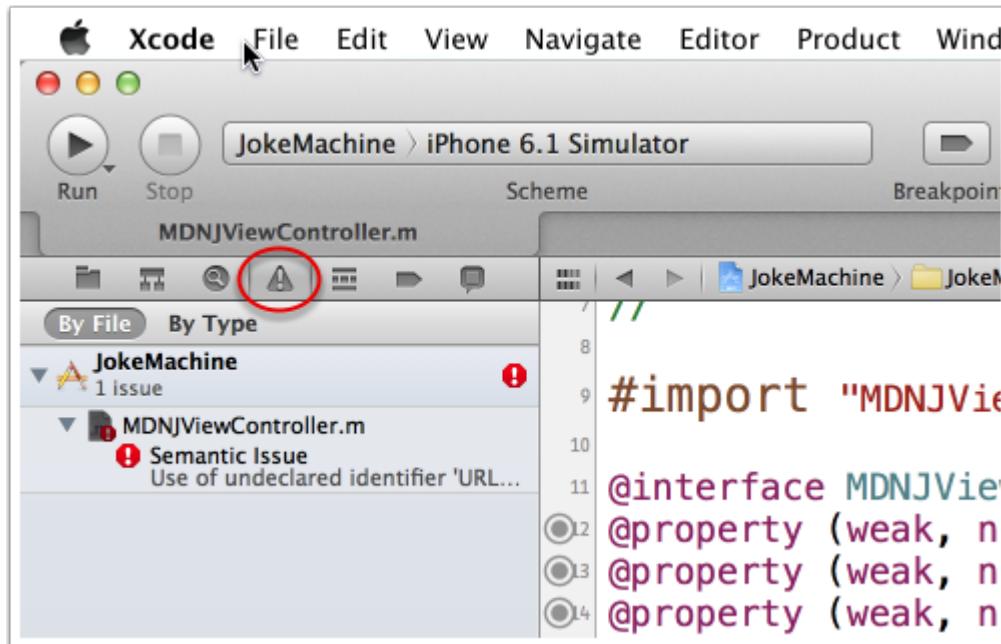


The screenshot shows the Xcode IDE interface. The top bar displays "DeleteME.xcodeproj — MDNJViewController.m" and "Finished running DeleteME on iPhone 6.1 Simulator". The bottom status bar shows "Project 1". The code editor window contains the following code:

```
Simulator Breakpoints Project 1
DeleteME > DeleteME > MDNJViewController.m > -[actionGetJokes:]  
- (IBAction)actionGetJokes:(id)sender {  
    // Get joke and set text View  
  
    NSURL *url = [NSURL URLWithString:@"http://api.icndb.com/jokes/random"];  
    NSURLRequest *urlRequest = [NSURLRequest requestWithURL:url];
```

A red exclamation mark icon is visible in the gutter of line 5, indicating a warning.

Warning Panel



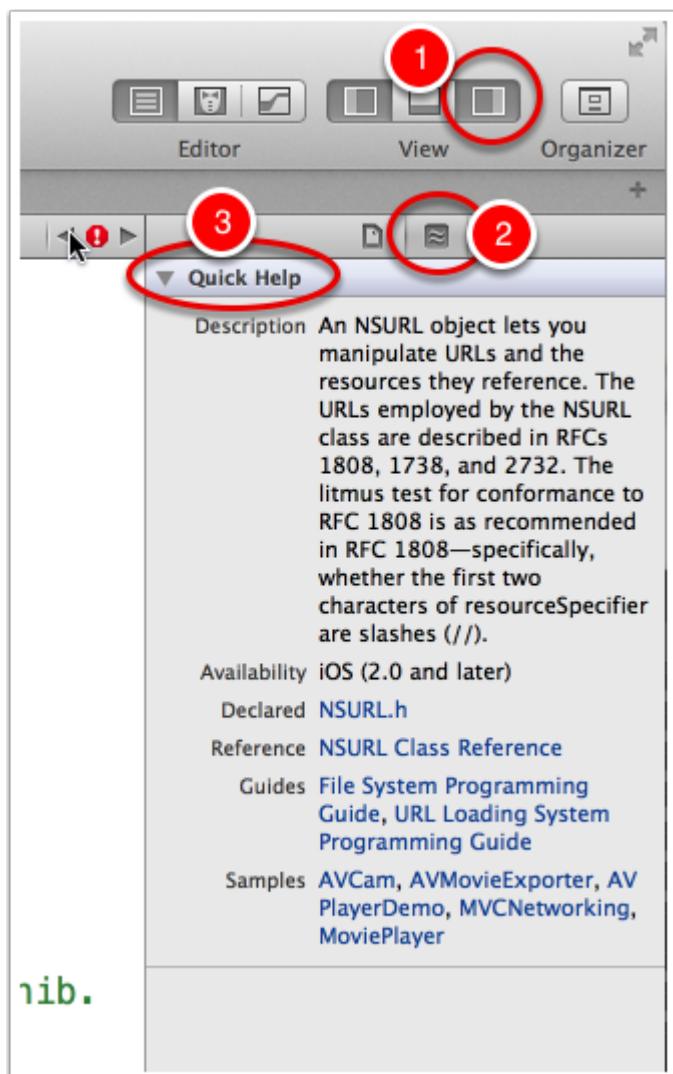
The screenshot shows the Xcode IDE interface with the "Warning Panel" open. The menu bar includes "Xcode", "File", "Edit", "View", "Navigate", "Editor", "Product", and "Window". The toolbar has "Run" and "Stop" buttons. The "Scheme" dropdown is set to "iPhone 6.1 Simulator". The "Breakpoint" button is also present. The "MDNJViewController.m" file is selected in the project navigation pane. The warning panel shows a list of issues:

- JokeMachine: 1 issue (circled)
- MDNJViewController.m: Semantic Issue (circled)
Use of undeclared identifier 'URL...'

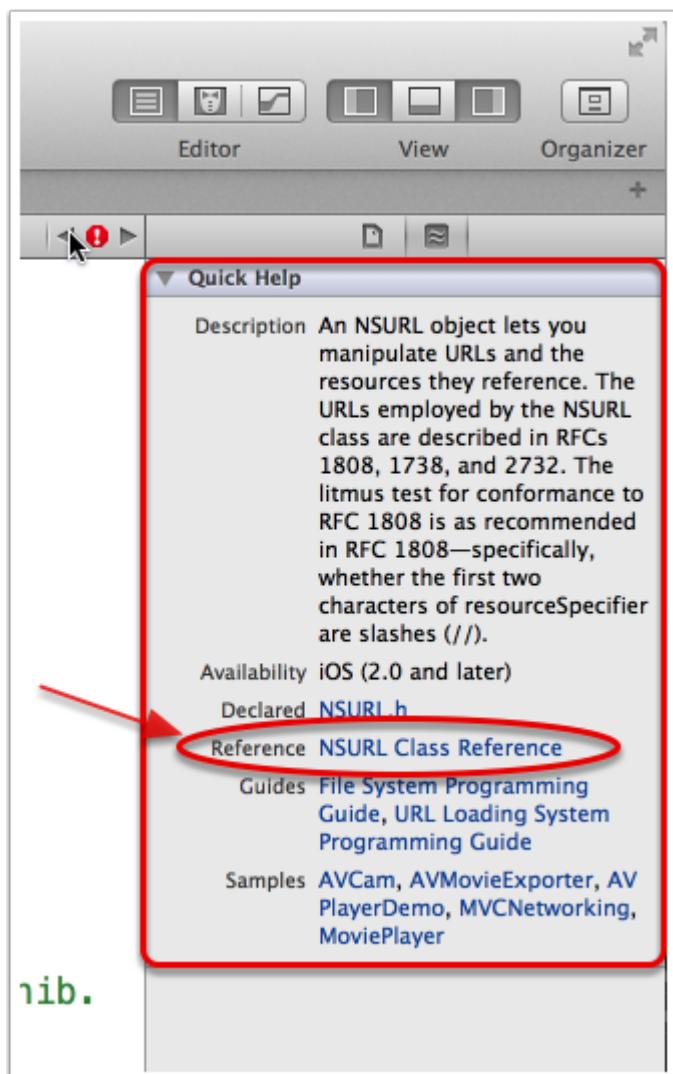
The code editor shows the following code with a semantic issue:

```
//  
#import "MDNJViewContro...  
  
@interface MDNJViewContro...  
@property (weak, n...  
@property (weak, n...  
@property (weak, n...
```

Xcode - Help While Editing



Xcode - Help While Editing



Organizer - Documentation

NSURLWithString:

Creates and returns an NSURL object initialized with a provided string.

+ (id)URLWithString:(NSString *)URLString

Parameters

URLString

The string with which to initialize the NSURL object. Must be a URL that conforms to RFC 2396. This method parses URLString according to RFCs 1738 and 1808. (To create NSURL objects for file system paths, use `fileURLWithPath:isDirectory:` instead.)

Return Value

An NSURL object initialized with `URLString`. If the string was malformed, returns `nil`.



Discussion

This method expects `URLString` to contain any necessary percent escape codes, which are ':', '/', '%', '#', ';', and '@'. Note that '%' escapes are translated via UTF-8.

Availability

Available in iOS 2.0 and later.

Related Sample Code

[AdvancedURLConnections](#)

[LazyTableImages](#)

JokeMachine.xcodeproj — MDNJViewController.m

```
NSURL *url = [NSURL URLWithString:@"http://api.icndb.com/jokes/random"];
```

JokeMachine.xcodeproj — MDNJViewController.m

```
NSURL *url = [NSURL URLWithString:@"http://api.icndb.com/jokes/random"];
```

We're going to do this instead!

```
NSString * urlString = @"http://api.icndb.com/jokes/random";
NSURL *url = [NSURL URLWithString:urlString];
```

JokeMachine.xcodeproj — MDNJViewController.m

```
i3 - (IBAction)actionGetJokes:(id)sender {  
44  
45     NSString * urlString = @"http://api.icndb.com/jokes/random";  
46     NSArray * nameArray = [NSArray array];  
47     NSError * error;  
48}
```

JokeMachine.xcodeproj — MDNJViewController.m

```
if (_myName.length > 0) {  
    nameArray = [_myName componentsSeparatedByString:@" "];  
    urlString = [urlString stringByAppendingFormat:@"?firstName=%@",  
                 nameArray[0]];  
  
    if ([nameArray count] > 1) {  
        urlString = [urlString stringByAppendingFormat:@"&lastName=%@",  
                     [nameArray lastObject]];  
    }  
}
```

JokeMachine.xcodeproj — MDNJViewController.m

```
59     NSURL * url = [NSURL URLWithString:urlString];  
60  
61     NSURLRequest * urlRequest = [NSURLRequest requestWithURL:url];  
62
```

JokeMachine.xcodeproj — MDNJViewController.m

```
// Will hold response from Server  
NSURLResponse * urlResponse;  
// Call Joke API  
NSData * data = [NSURLConnection sendSynchronousRequest:urlRequest  
                                                 returningResponse:&urlResponse error:&error];
```

JokeMachine.xcodeproj — MDNJViewController.m

```
63
64     NSURLResponse *urlResponse;
65     NSData *data = [NSURLConnection sendSynchronousRequest:urlRequest
66         returningResponse:&urlResponse error:&error];
67
68     if (!error) {
69 }
```

JokeMachine.xcodeproj — MDNJViewController.m

```
if (!error) {
} else {
    self.txtView.text = @"No Joke For You!";
}
```

JokeMachine.xcodeproj — MDNJViewController.m

```
NSDictionary* json = [NSJSONSerialization JSONObjectWithData:data
options:kNilOptions error:&error];
```

JokeMachine.xcodeproj — MDNJViewController.m

```
NSDictionary* json = [NSJSONSerialization JSONObjectWithData:data
options:kNilOptions error:&error];
NSString *jsonString = [[json objectForKey:@"value"]
objectForKey:@"joke"];
```

JokeMachine.xcodeproj — MDNJViewController.m

Joke is in Categories, which is in Value

```
All Output ▾ Printing description of json:
{
    type = success;
    value =     {
        categories =         (
        );
        id = 399;
        joke = "In the movie "The Matrix";,
Chuck Norris is the Matrix. If you pay close
attention in the green "falling code";
scenes, you can make out the faint texture of his
beard.";
    };
}
```

String Object instead of JSON

What if Server returned a String instead of a JSON Object?

```
NSString *stringData = [NSString string] (hit Esc key)
id string
id stringWithCharacters:(const unichar *) length:(NSUInteger)
id stringWithContentsOfFile:(NSString *)
id stringWithContentsOfFile:(NSString *) encoding:(NSStringEncoding) error:
id stringWithContentsOfFile:(NSString *) usedEncoding:(NSStringEncoding)
id stringWithContentsOfURL:(NSURL *)
id stringWithContentsOfURL:(NSURL *) encoding:(NSStringEncoding) error:
id stringWithContentsOfURL:(NSURL *) usedEncoding:(NSStringEncoding *)
returns an empty string. More...
```

String Object instead of JSON

What if Server returned a String instead of a JSON Object?

```
self.txtView.text = [NSString stringWithContentsOfURL:url encoding: NSUTF8StringEncoding error:&error];
```

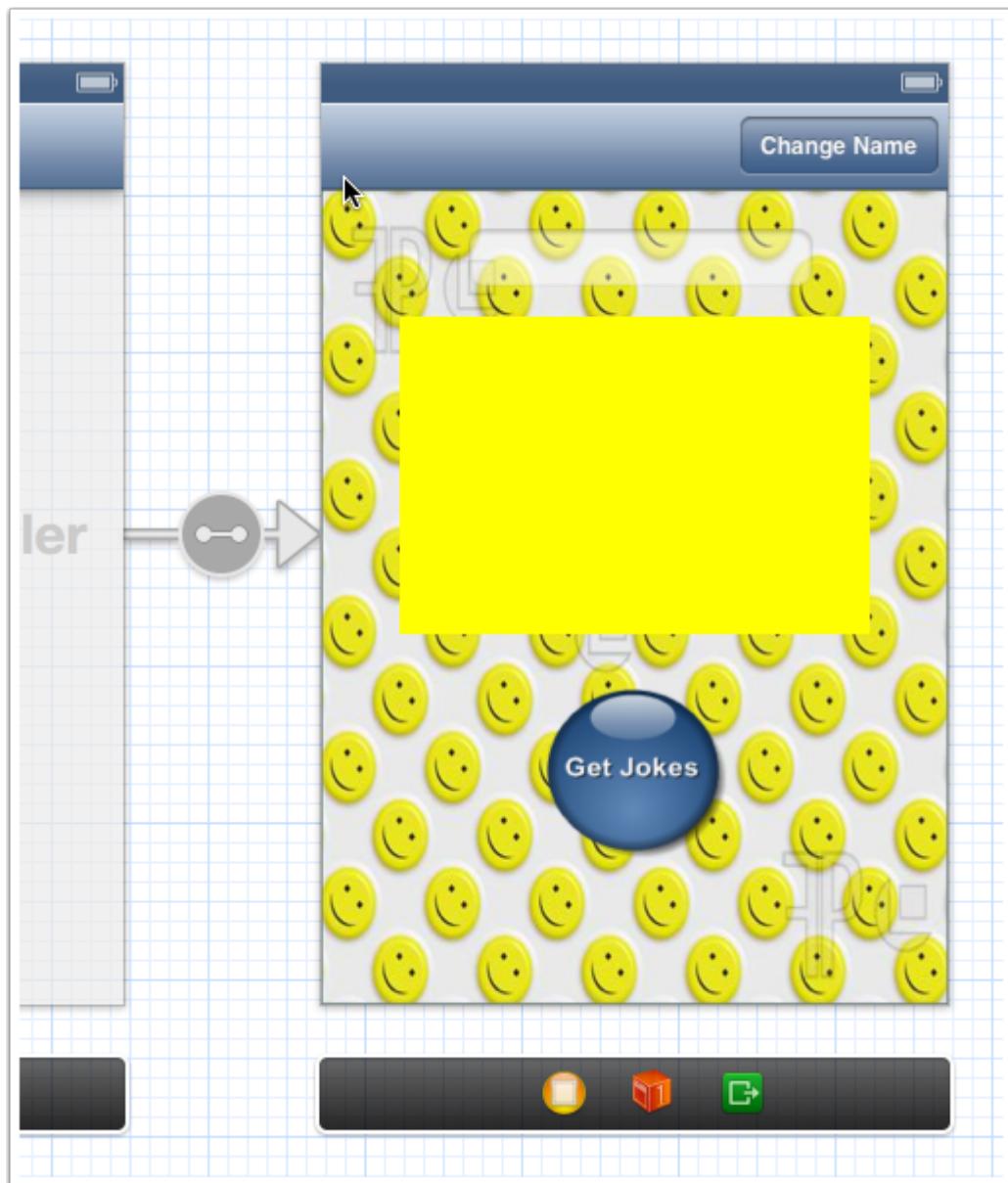
JokeMachine.xcodeproj — MDNJViewController.m

```
NSDictionary* json = [NSJSONSerialization JSONObjectWithData:data options:kNilOptions error:&error];
NSString *jsonString = [[json objectForKey:@"value"]
                        objectForKey:@"joke"];
self.txtView.text = jsonString;
```

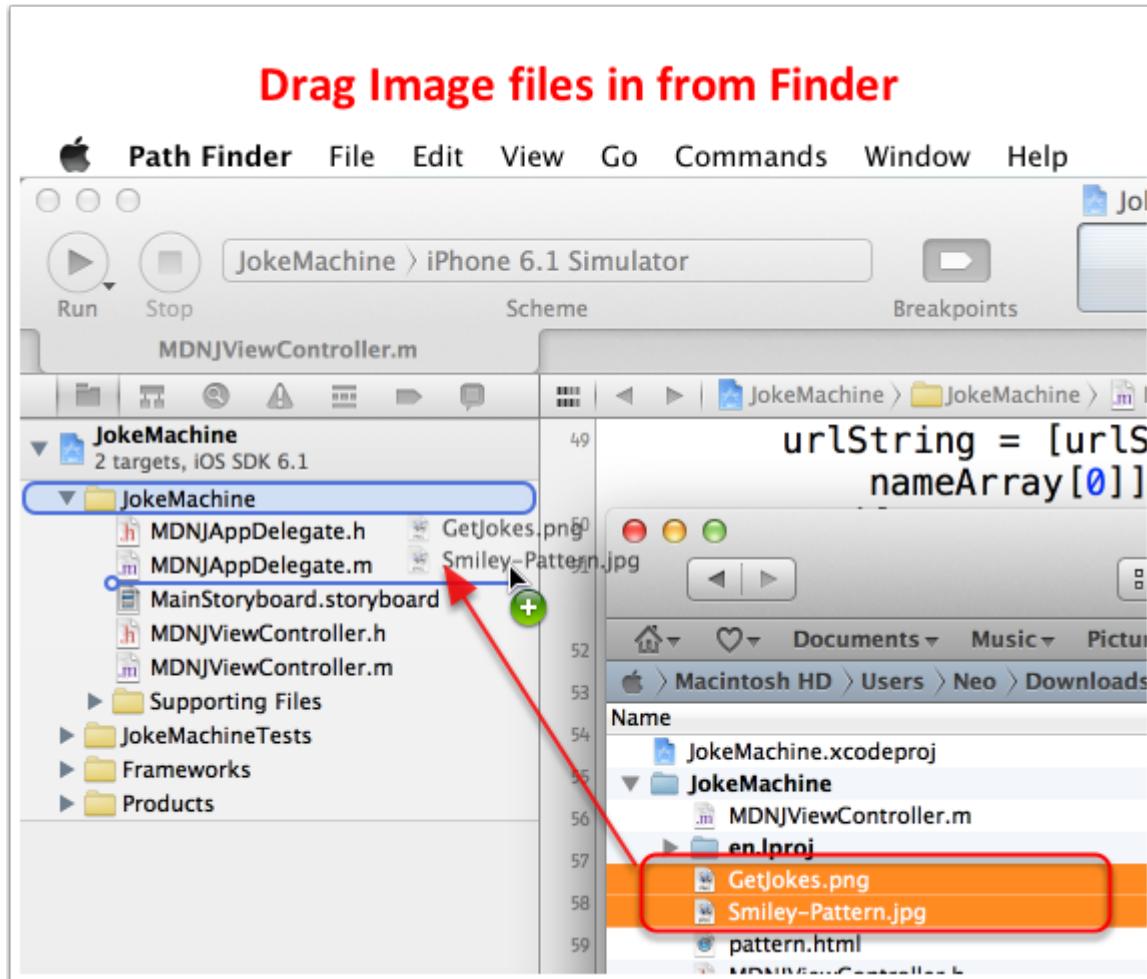
JokeMachine.xcodeproj — MDNJViewController.m

```
43 - (IBAction)actionGetJokes:(id)sender {
44     NSString * urlString = @"http://api.icndb.com/jokes/random";
45     NSArray * nameArray = [NSArray array];
46     NSError * error;
47     if (_myName.length > 0) {
48         nameArray = [_myName componentsSeparatedByString:@" "];
49         urlString = [urlString stringByAppendingFormat:@"?firstName=%@",
50                     nameArray[0]];
51         if ([nameArray count] > 1) {
52             urlString = [urlString stringByAppendingFormat:@"&lastName=%@",
53                         [nameArray lastObject]];
54         }
55     }
56     NSURL * url = [NSURL URLWithString:urlString];
57     NSURLRequest * urlRequest = [NSURLRequest requestWithURL:url];
58     NSURLResponse * urlResponse;
59     NSData * data = [NSURLConnection sendSynchronousRequest:urlRequest
60                                         returningResponse:&urlResponse error:&error];
61     if (!error) {
62         NSDictionary* json = [NSJSONSerialization JSONObjectWithData:data
63                                         options:kNilOptions error:&error];
64         NSString * jsonString = [[json objectForKey:@"value"]
65                               objectForKey:@"joke"];
66         self.txtView.text = jsonString;
67     } else {
68         self.txtView.text = @"No Joke For You!";
69     }
70 }
```

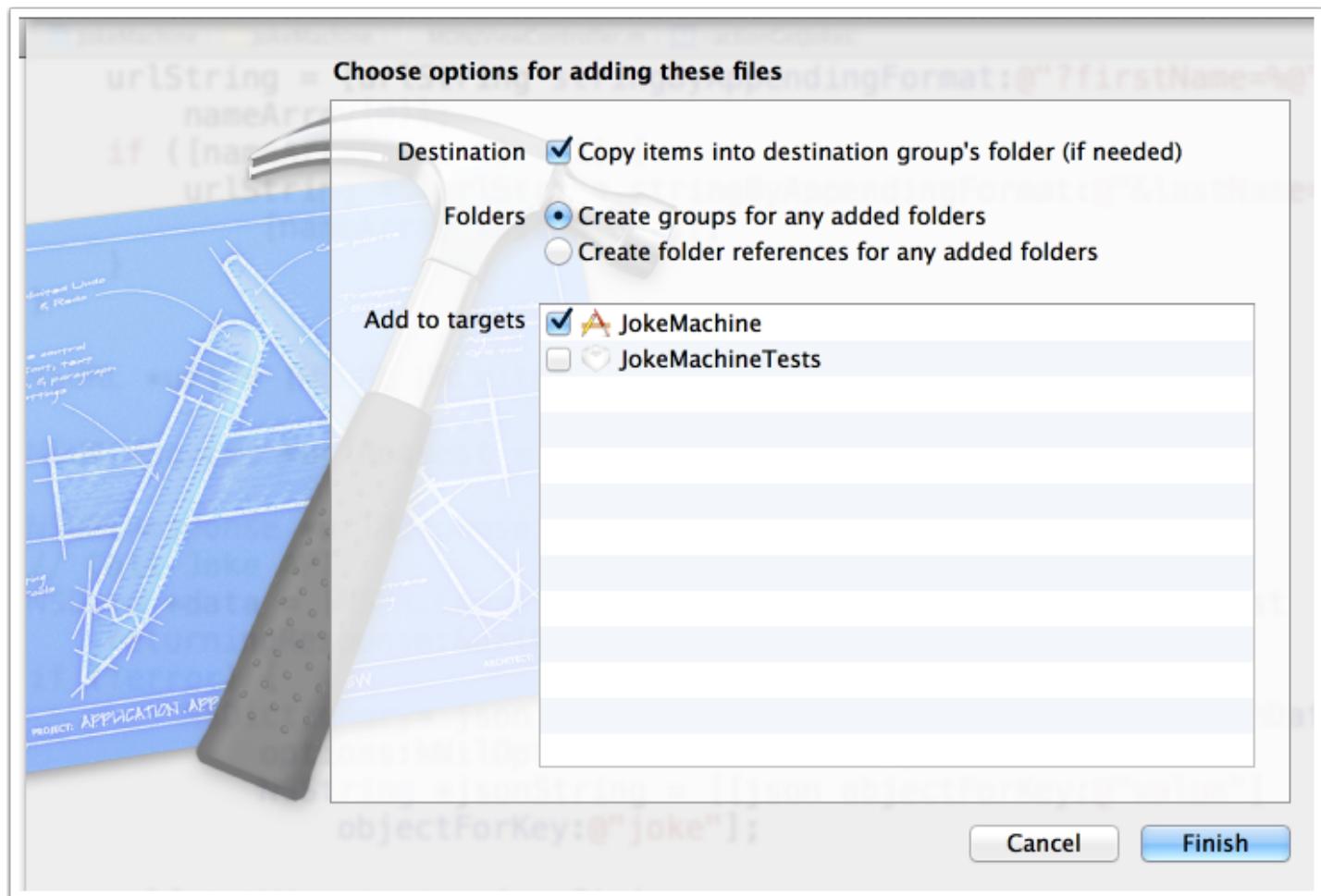
JokeMachine.xcodeproj — MainStoryboard.storyboard



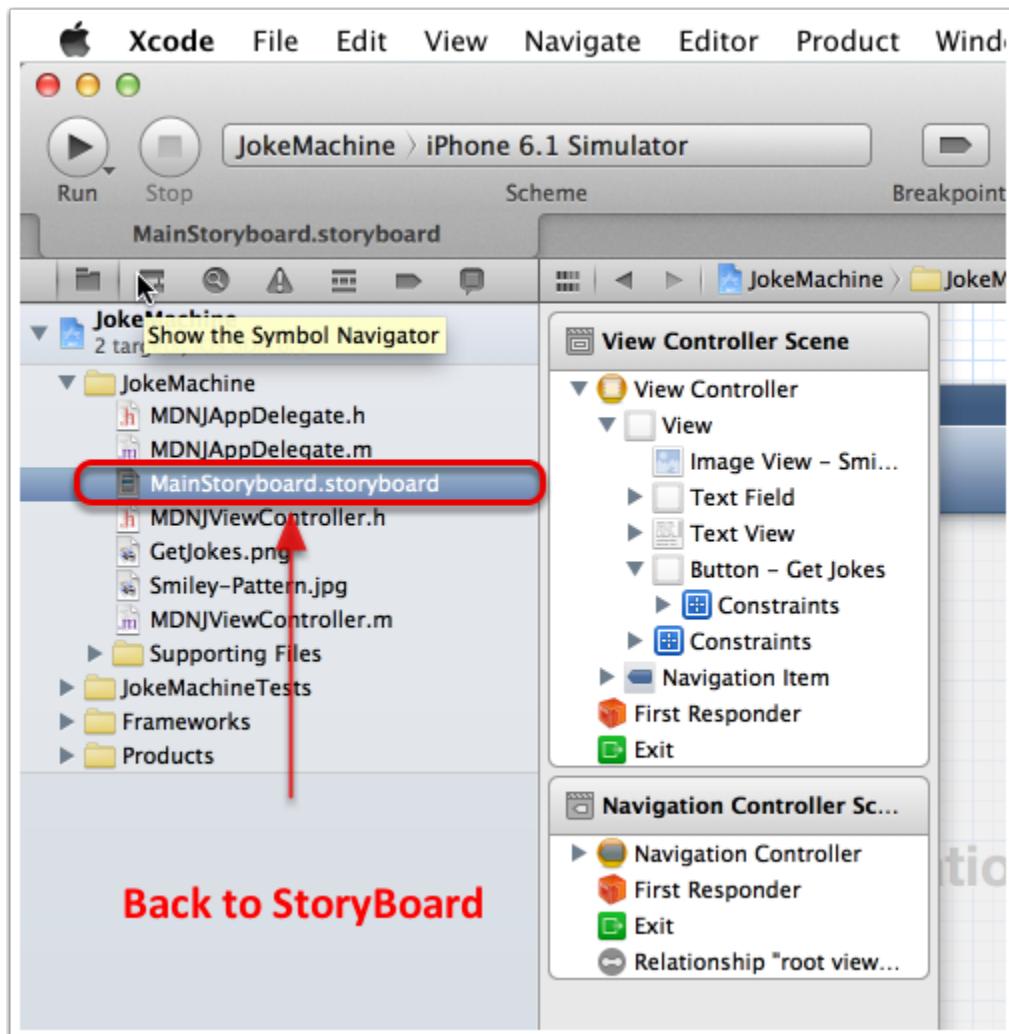
Path Finder



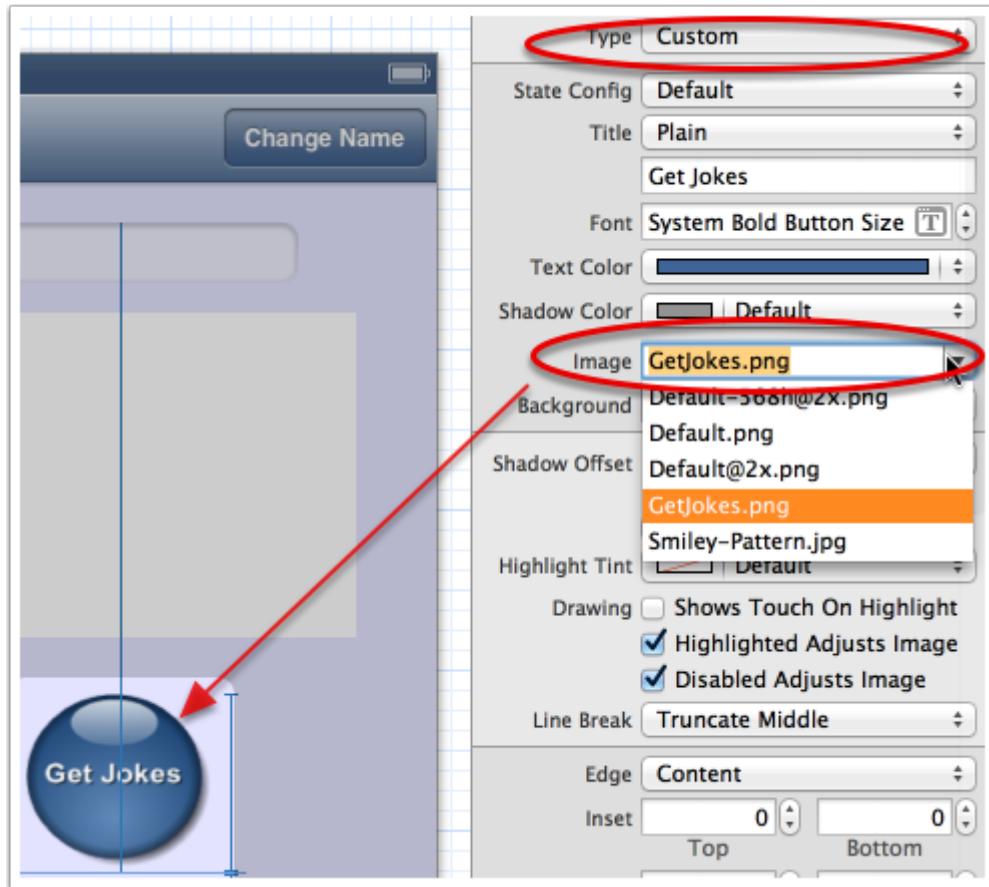
Xcode



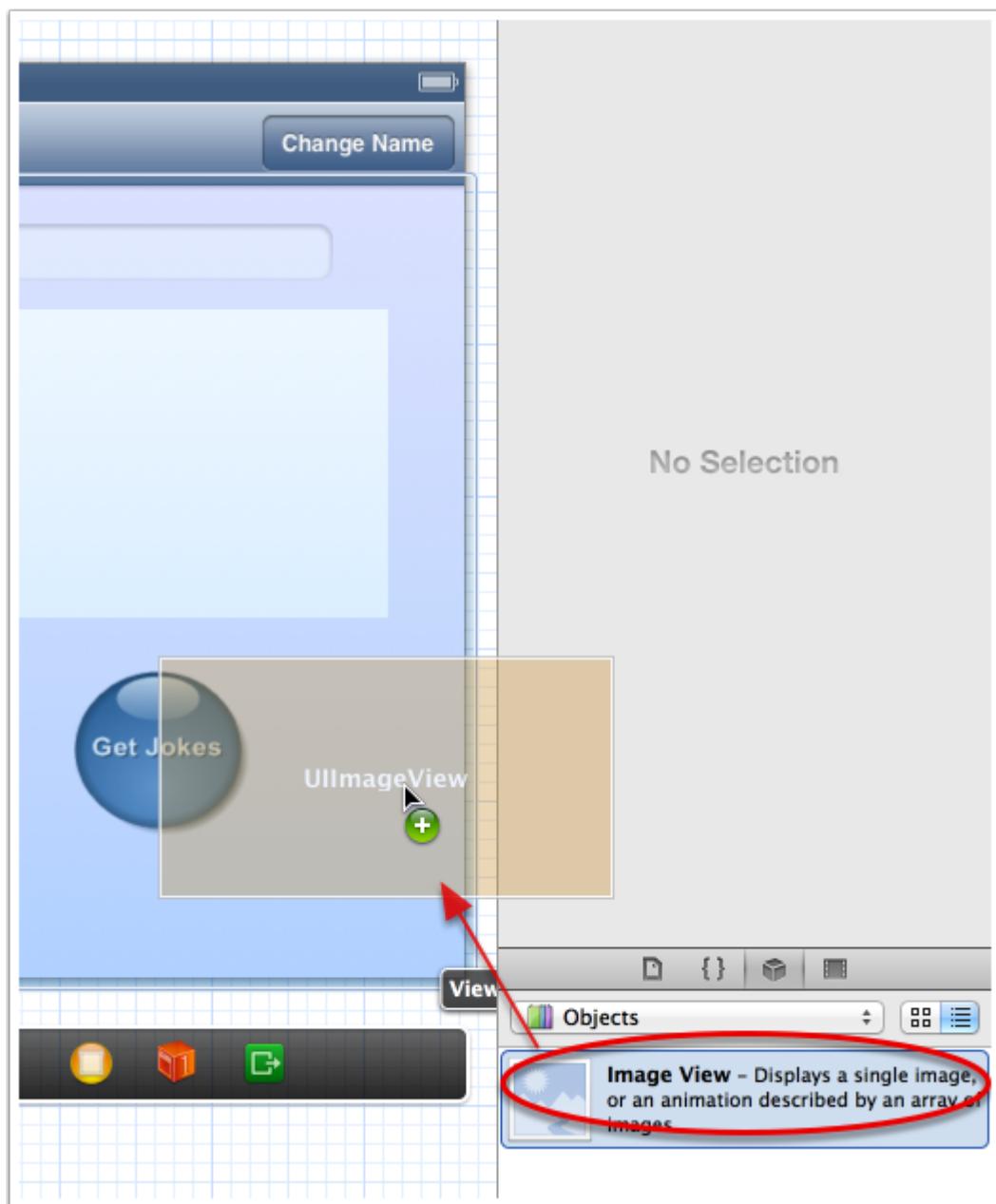
Xcode



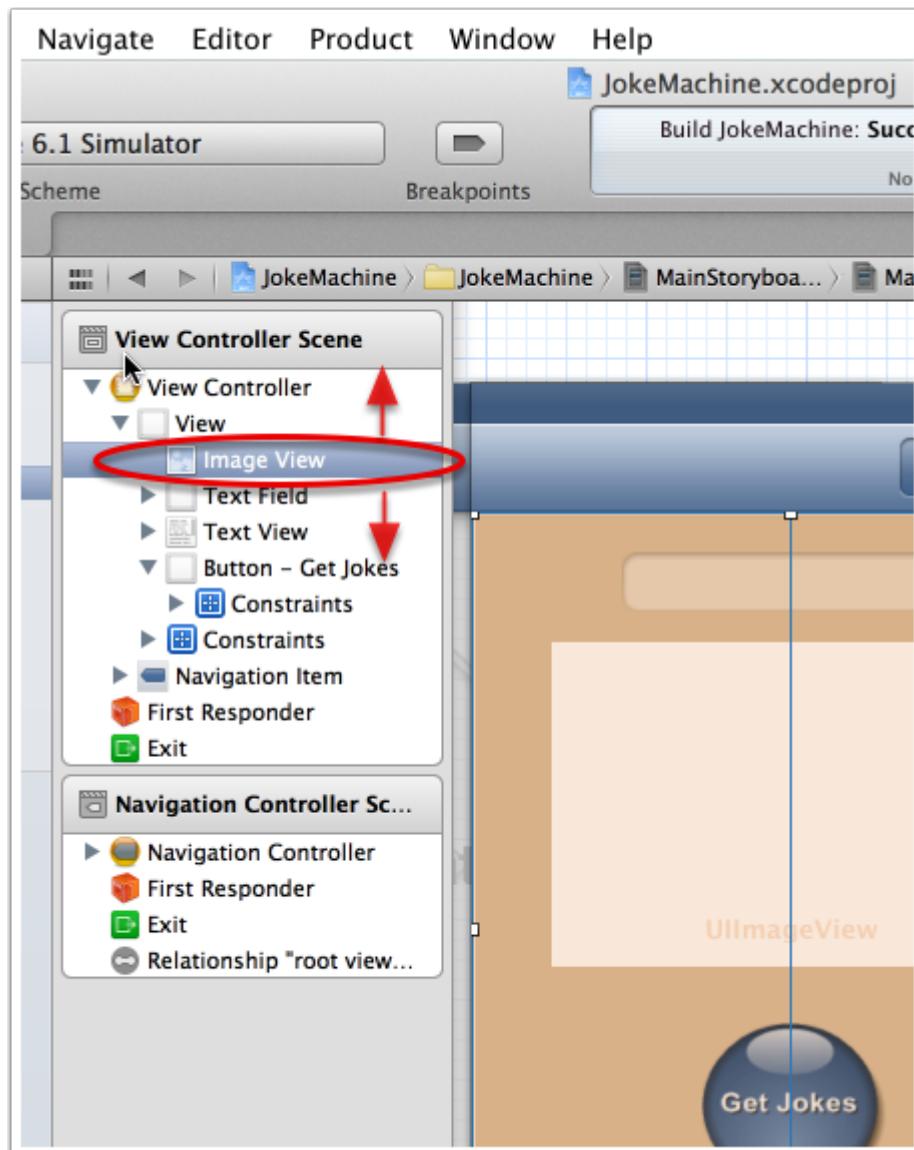
Add Image to UIButton using Custom Option



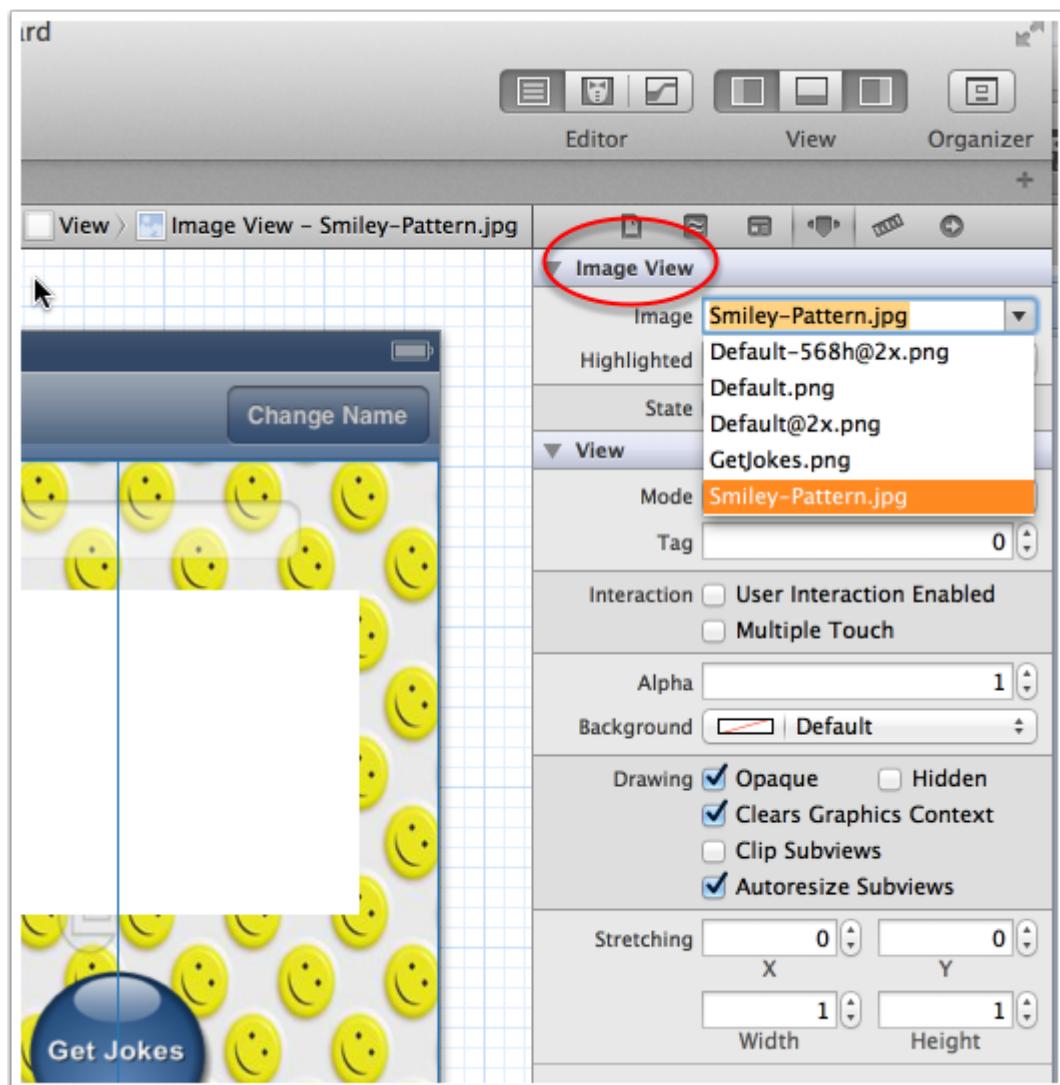
Add UIImageView Object to add Custom Background to UIView



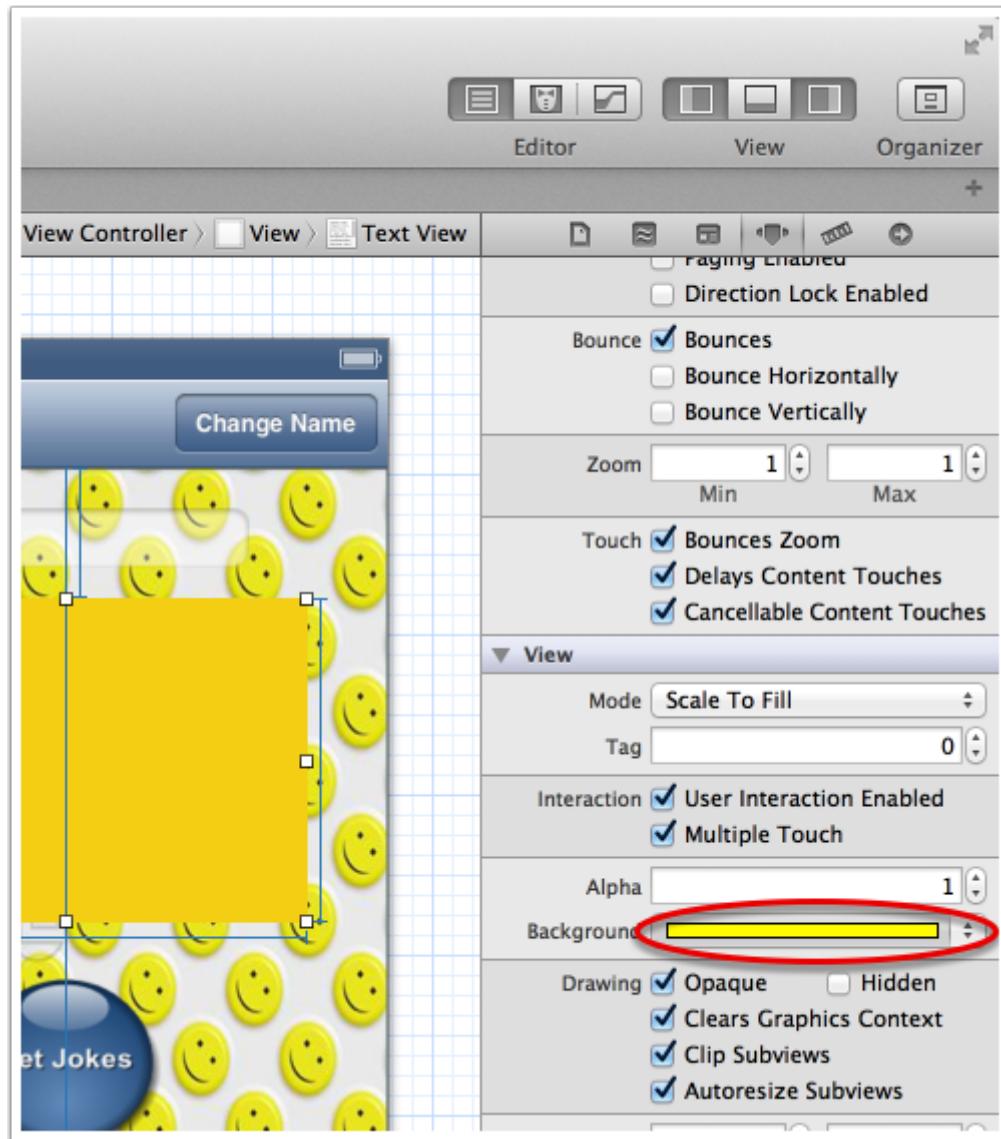
Positioning UIImageView



Select Background for UIView



Set Color



TaDa! Let's Build it for Real

Thank you for Attending! @MobileDevNJ