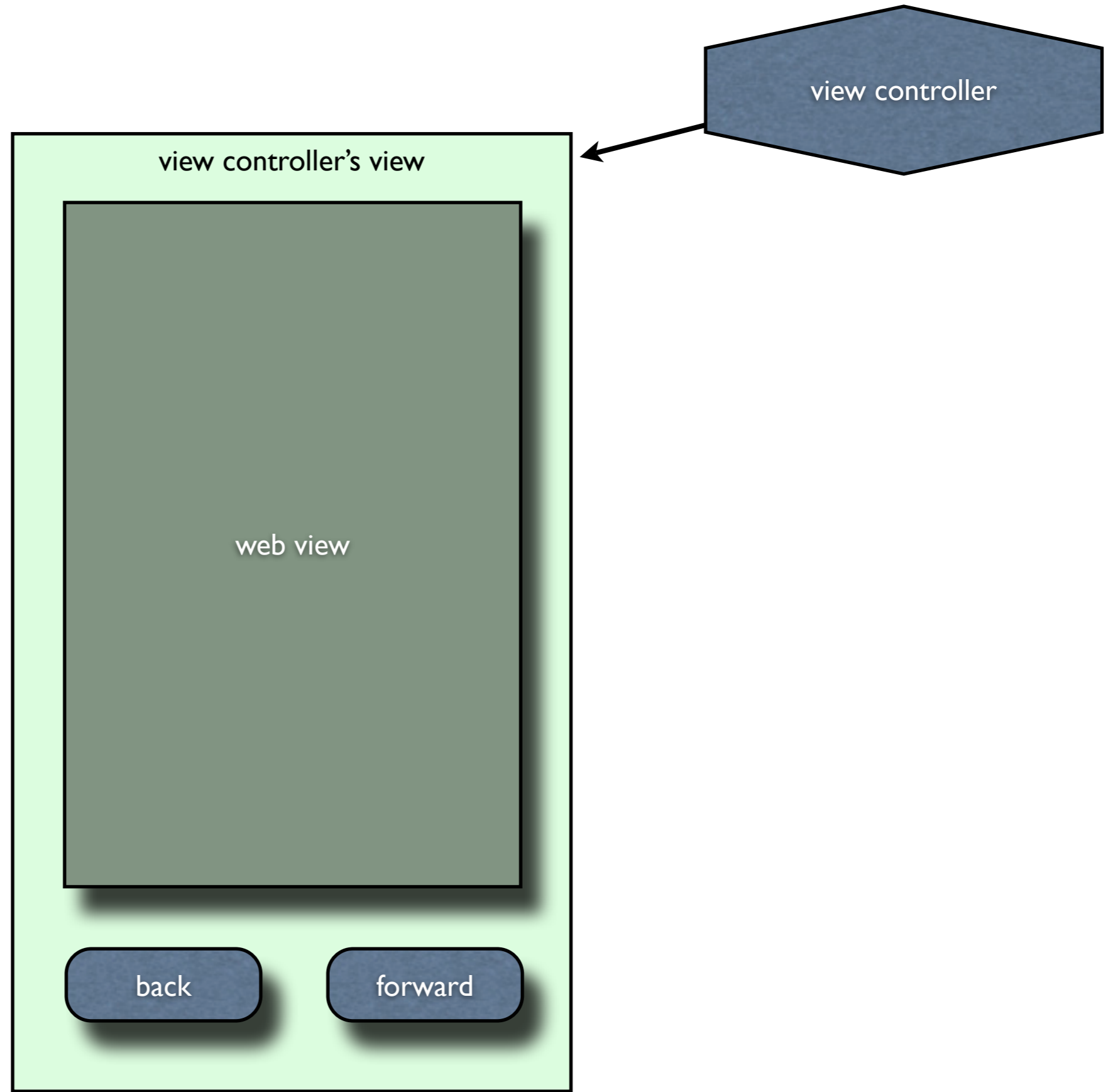
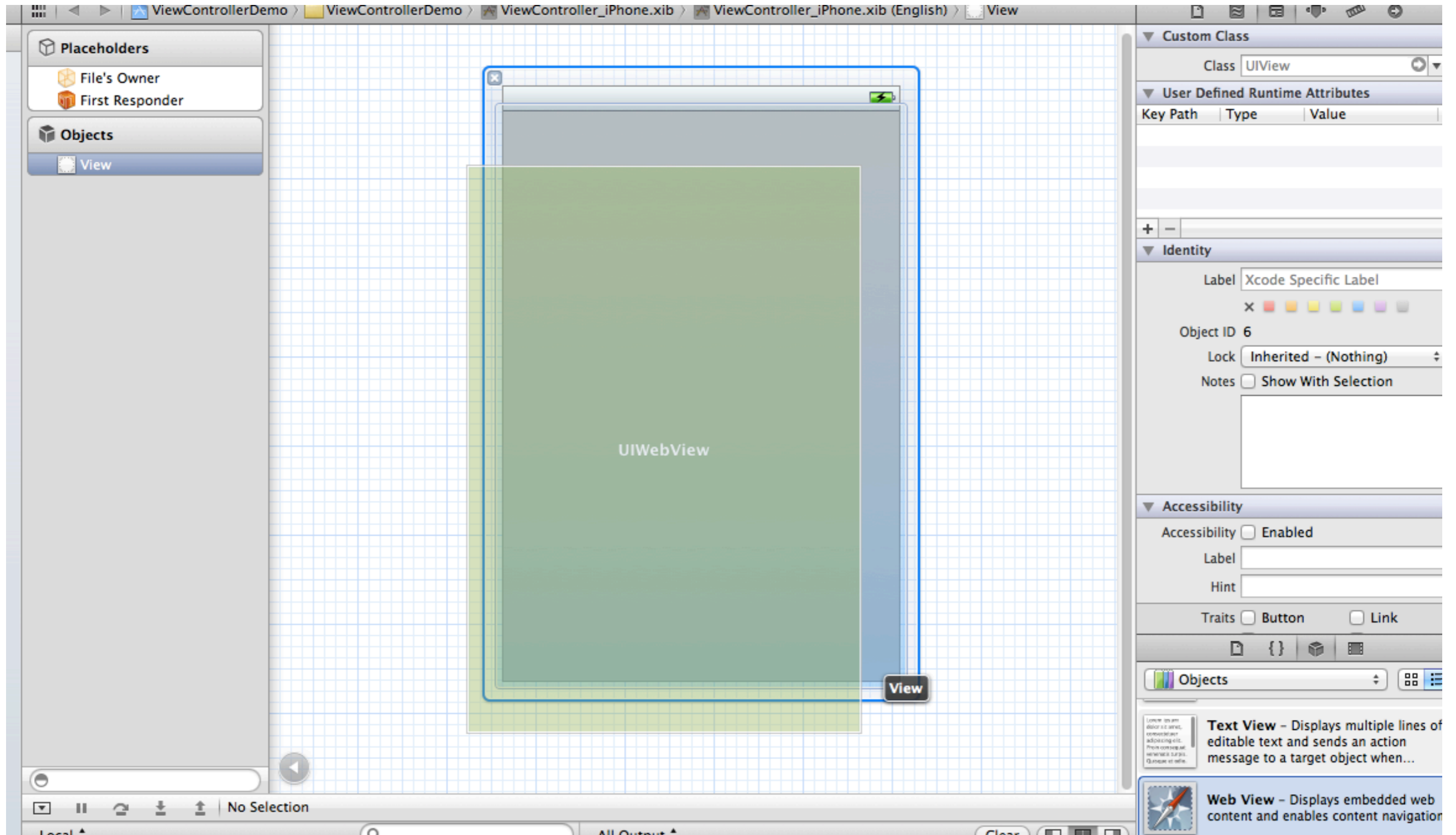


Designing views

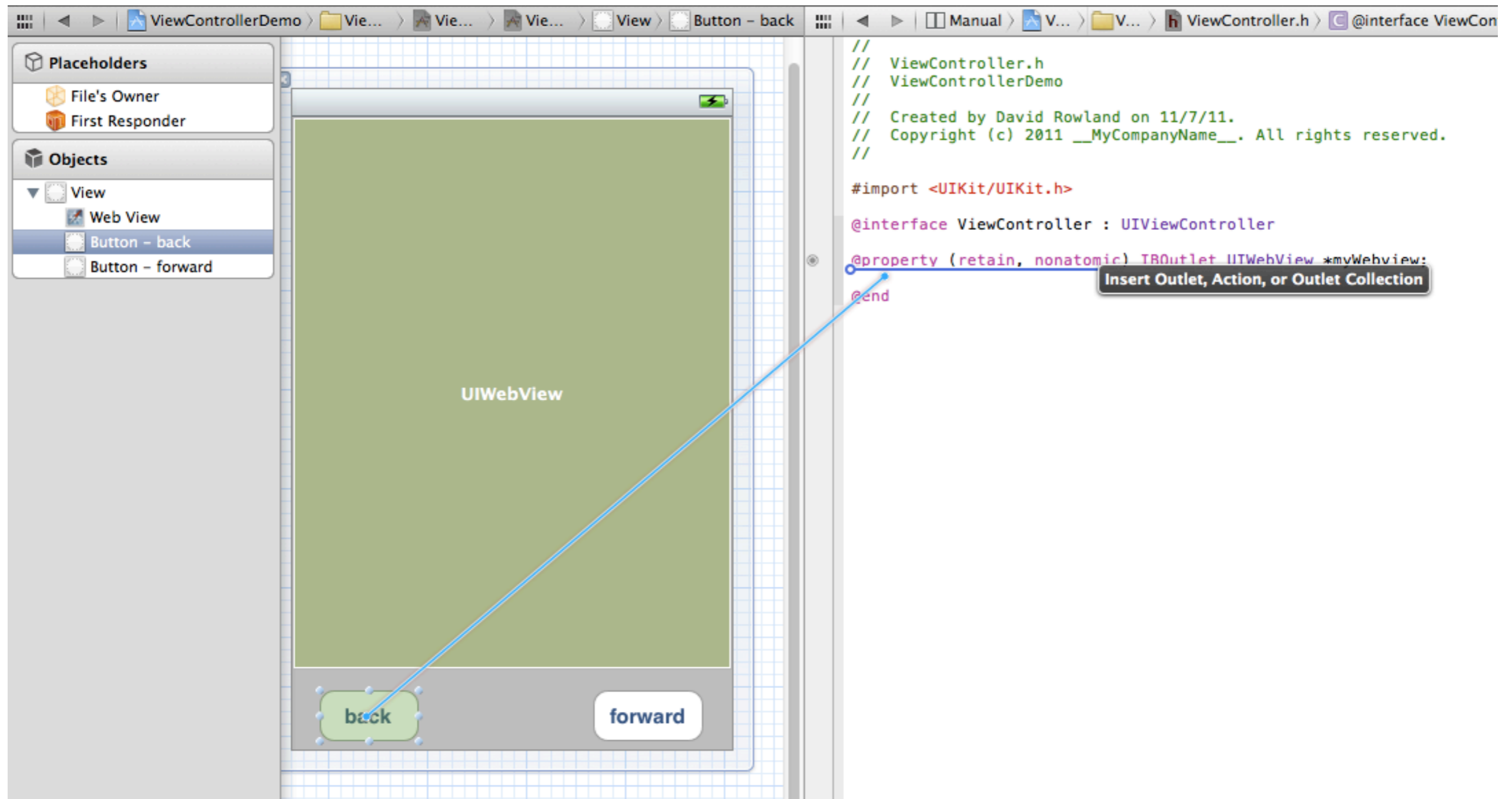


To use Interface Builder, create a new view controller file pair and ask for an xib file as well. Drag a webview from the library onto the view. Locate and size it.



Drag and locate two buttons. Double click each to enter its title.

ctrl-drag from the webview onto the @interface of the view controller to make IB write its definition. It will make a @property declaration, and in the @implementation it will @synthesize it. Do the same for the buttons.

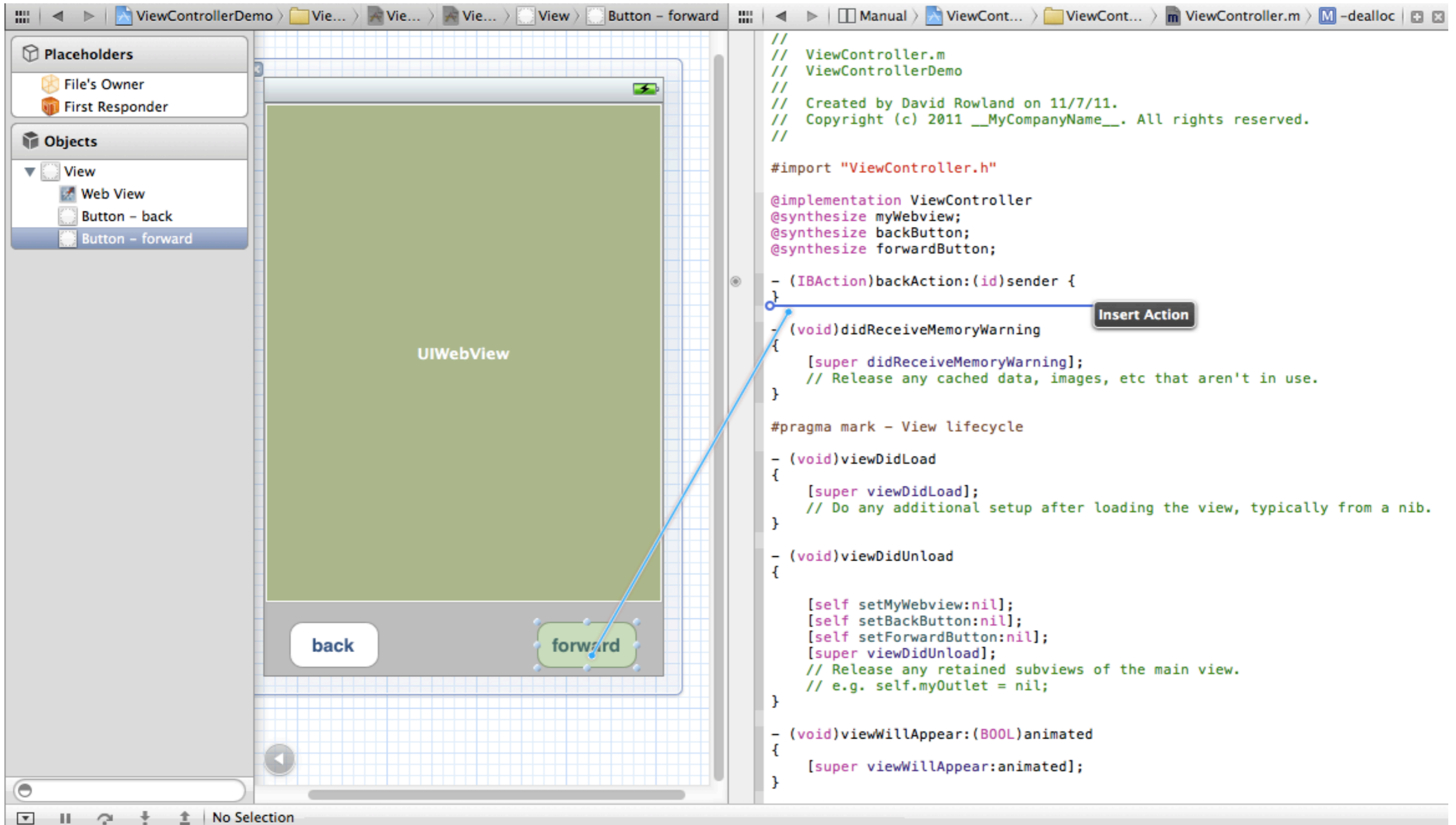


ctrl-drag from a button into the @implementation area.

The screenshot shows the Xcode interface with a storyboard on the left and the implementation file on the right. The storyboard displays a UIWebView with two buttons at the bottom: 'back' and 'forward'. The 'back' button is highlighted with a blue circle, and a blue arrow points from it to the @implementation block in the code. The code in ViewController.m includes the following:

```
//  
// ViewController.m  
// ViewControllerDemo  
//  
// Created by David Rowland on 11/7/11.  
// Copyright (c) 2011 __MyCompanyName__. All rights reserved.  
//  
#import "ViewController.h"  
  
@implementation ViewController  
@synthesize myWebview;  
@synthesize backButton;  
@synthesize forwardButton;  
  
- (void)didReceiveMemoryWarning  
{  
    [super didReceiveMemoryWarning];  
    // Release any cached data, images, etc that aren't in use.  
}  
  
#pragma mark - View lifecycle  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.  
}  
  
- (void)viewDidUnload  
{  
    [self setMyWebview:nil];  
    [self setBackButton:nil];  
    [self setForwardButton:nil];  
    [super viewDidUnload];  
    // Release any retained subviews of the main view.  
    // e.g. self.myOutlet = nil;  
}  
  
- (void)viewWillAppear:(BOOL)animated  
{  
    [super viewWillAppear:animated];  
}
```

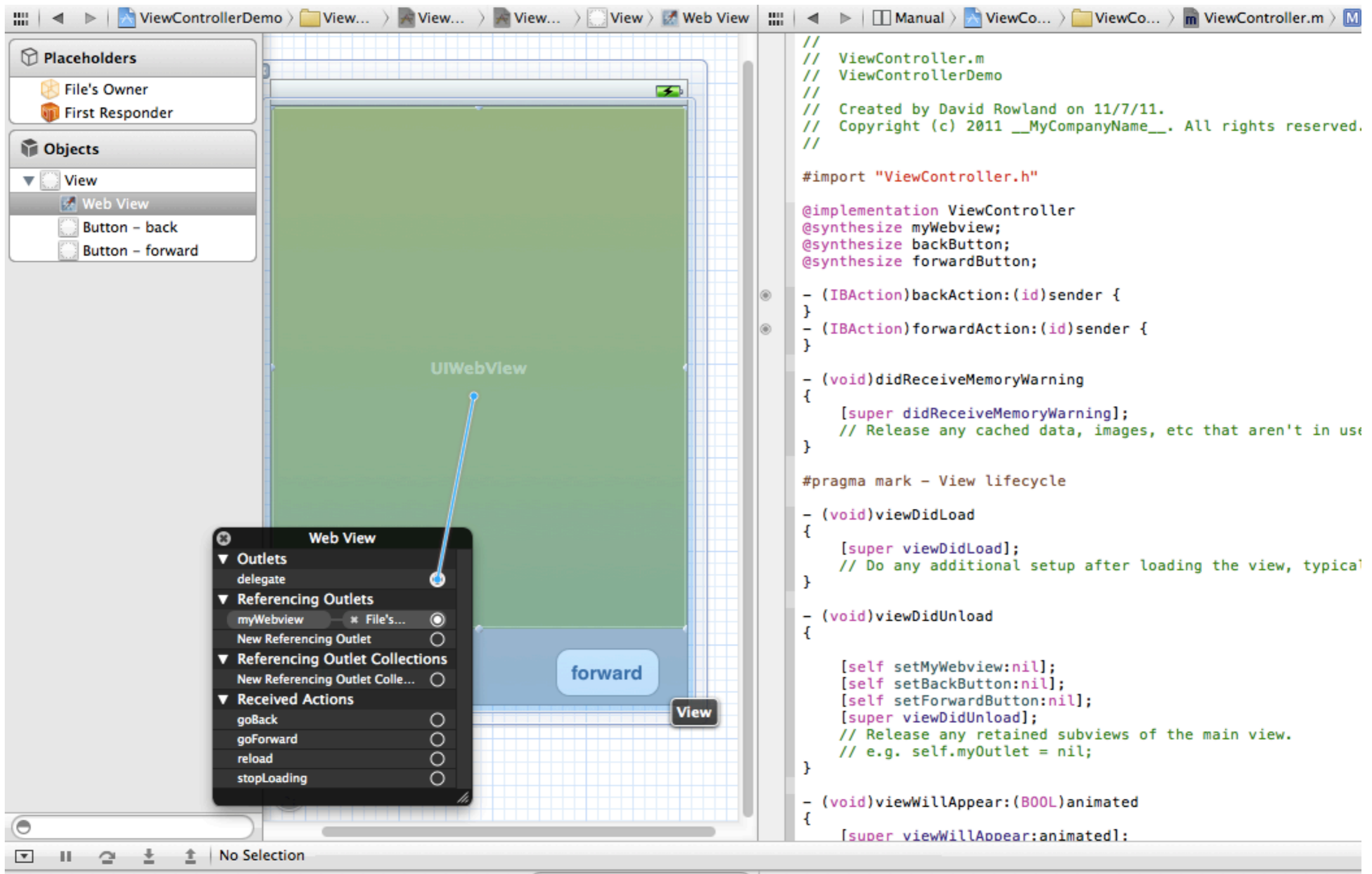
IB will write a skeleton of an action method and connect it to the button.



The screenshot displays the Xcode environment. On the left, the Interface Builder (IB) window shows a UIWebView with two buttons: 'back' and 'forward'. The 'forward' button is selected, and a blue line connects it to the 'backAction:' method in the code file on the right. A tooltip 'Insert Action' is visible over the code. The code file shows the implementation of the ViewController class, including the 'backAction:' method.

```
//  
// ViewController.m  
// ViewControllerDemo  
//  
// Created by David Rowland on 11/7/11.  
// Copyright (c) 2011 __MyCompanyName__. All rights reserved.  
//  
#import "ViewController.h"  
  
@implementation ViewController  
@synthesize myWebview;  
@synthesize backButton;  
@synthesize forwardButton;  
  
- (IBAction)backAction:(id)sender {  
}  
  
- (void)didReceiveMemoryWarning  
{  
    [super didReceiveMemoryWarning];  
    // Release any cached data, images, etc that aren't in use.  
}  
  
#pragma mark - View lifecycle  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.  
}  
  
- (void)viewDidUnload  
{  
  
    [self setMyWebview:nil];  
    [self setBackButton:nil];  
    [self setForwardButton:nil];  
    [super viewDidUnload];  
    // Release any retained subviews of the main view.  
    // e.g. self.myOutlet = nil;  
}  
  
- (void)viewWillAppear:(BOOL)animated  
{  
    [super viewWillAppear:animated];  
}
```

ctrl-click on the web view. A dialog will appear. Drag from the delegate outlet into the view. This will make the view controller the delegate for the webview. You must still write the delegate methods.



The screenshot shows the Xcode interface with a storyboard on the left and a code editor on the right. The storyboard displays a green `UIWebView` and a blue `forward` button. A context menu for the `Web View` is open, showing the `delegate` outlet selected. The code editor shows the implementation of the `ViewController` class, including the `delegate` outlet and several delegate methods.

```
//  
// ViewController.m  
// ViewControllerDemo  
//  
// Created by David Rowland on 11/7/11.  
// Copyright (c) 2011 __MyCompanyName__. All rights reserved.  
//  
  
#import "ViewController.h"  
  
@implementation ViewController  
@synthesize myWebview;  
@synthesize backButton;  
@synthesize forwardButton;  
  
- (IBAction)backAction:(id)sender {  
}  
- (IBAction)forwardAction:(id)sender {  
}  
  
- (void)didReceiveMemoryWarning  
{  
    [super didReceiveMemoryWarning];  
    // Release any cached data, images, etc that aren't in use  
}  
  
#pragma mark - View lifecycle  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typical  
}  
  
- (void)viewDidUnload  
{  
    [self setMyWebview:nil];  
    [self setBackButton:nil];  
    [self setForwardButton:nil];  
    [super viewDidUnload];  
    // Release any retained subviews of the main view.  
    // e.g. self.myOutlet = nil;  
}  
  
- (void)viewWillAppear:(BOOL)animated  
{  
    [super viewWillAppear:animated];
```

If the button actions will simply call methods of the webview directly and do nothing else you need not create their skeletons. You can drag from the dialog onto the button.

The screenshot displays the Xcode interface for a storyboard. On the left, the 'Objects' palette shows a 'Web View' and two buttons: 'Button - back' and 'Button - forward'. The main canvas shows a 'UIWebView' with a green background and two buttons at the bottom: 'back' and 'forward'. A 'Web View' dialog is open, showing a list of actions under 'Received Actions'. A blue arrow points from the 'goForward' action to the 'forward' button. The right side of the image shows the Objective-C code for 'ViewController.m'.

```
//  
// ViewController.m  
// ViewControllerDemo  
//  
// Created by David Rowland on 11/7/11.  
// Copyright (c) 2011 __MyCompanyName__. All rights reserved.  
//  
  
#import "ViewController.h"  
  
@implementation ViewController  
@synthesize myWebview;  
@synthesize backButton;  
@synthesize forwardButton;  
  
- (IBAction)backAction:(id)sender {  
}  
- (IBAction)forwardAction:(id)sender {  
}  
  
- (void)didReceiveMemoryWarning  
{  
    [super didReceiveMemoryWarning];  
    // Release any cached data, images, etc that aren't in use  
}  
  
#pragma mark - View lifecycle  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically  
}  
  
- (void)viewDidUnload  
{  
  
    [self setMyWebview:nil];  
    [self setBackButton:nil];  
    [self setForwardButton:nil];  
    [super viewDidUnload];  
    // Release any retained subviews of the main view.  
    // e.g. self.myOutlet = nil;  
}  
  
- (void)viewWillAppear:(BOOL)animated  
{  
    [super viewWillAppear:animated];  
}
```

That will establish the action connection without writing any explicit code.

The screenshot displays the Xcode interface for a storyboard. On the left, the 'Objects' library shows a 'Web View' and two buttons labeled 'back' and 'forward'. The main canvas shows a green 'UIWebView' and the two buttons. A 'Web View' outlet is connected to the UIWebView. The 'Received Actions' list shows 'goBack' connected to the 'back' button and 'goForward' connected to the 'forward' button. A context menu is open over the 'back' button, showing various touch events, with 'Touch Up Inside' selected. The code editor on the right shows the implementation of the ViewController.m file.

```
//  
// ViewController.m  
// ViewControllerDemo  
//  
// Created by David Rowland on 11/7/11.  
// Copyright (c) 2011 __MyCompanyName__. All rights reserved.  
//  
  
#import "ViewController.h"  
  
@implementation ViewController  
@synthesize myWebview;  
@synthesize backButton;  
@synthesize forwardButton;  
  
- (IBAction)backAction:(id)sender {  
}  
- (IBAction)forwardAction:(id)sender {  
}  
  
- (void)didReceiveMemoryWarning  
{  
    [super didReceiveMemoryWarning];  
    // Release any cached data, images, etc that aren'  
}  
  
#pragma mark - View lifecycle  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view,  
}  
  
- (void)viewDidUnload  
{  
  
    [self setMyWebview:nil];  
    [self setBackButton:nil];  
    [self setForwardButton:nil];  
    [super viewDidUnload];  
    // Release any retained subviews of the main view.  
    // e.g. self.myOutlet = nil;  
}  
  
- (void)viewWillAppear:(BOOL)animated  
{  
    [super viewWillAppear:animated];
```

To accomplish these things in code you must:

Define them in the @interface of the view controller

```
@interface ViewController : UIViewController

@property (retain, nonatomic) UIWebView *myWebview;
@property (retain, nonatomic) UIButton *backButton;
@property (retain, nonatomic) UIButton *forwardButton;

@end
```

Synthesize them in the @implementation

```
@implementation ViewController
@synthesize myWebview;
@synthesize backButton;
@synthesize forwardButton;
```

Initialize them in viewDidLoad

```
CGRect screenRect = [[UIScreen mainScreen] bounds];
webView = [[UIWebView alloc] initWithFrame:screenRect]; //standard alloc, retain = 1
[webView setBackgroundColor:[UIColor clearColor]];
[webView setScalesPageToFit:YES];
[webView setDelegate:self];
NSString * urlString = @"http://apple.com";
NSURL *url = [NSURL URLWithString:urlString];
NSURLRequest *request = [NSURLRequest requestWithURL:url];
[webView loadRequest:request];
[self.view addSubview:webView];
[webView release];

CGRect buttonFrame = CGRectMake(12, 372, 54, 44);
backButton = [UIButton buttonWithType:UIButtonTypeRoundedRect]; //factory method, autoreleased, not retained
[backButton setFrame:buttonFrame];
[backButton setTitle:@"Back" forState:UIControlStateNormal];
[backButton addTarget:self action:@selector(backAction:) forControlEvents:UIControlEventTouchUpInside];
[self.view addSubview:backButton];

//similar for forward button
```

Write the button action methods

```
- (void)backAction:(id)sender
{
    [webView goBack];
}
```

Write the UIWebView delegate method

```
- (void)webViewDidFinishLoad:(UIWebView *)webView
{
    //Test whether pages exist forward or back and enable the buttons accordingly.
}
```

Interface Builder cannot write the delegate method. You must do that.
Also, some initialization, like the webview loadRequest must be done in code.

Oh, and one more thing.

The view controller must be defined and created somewhere. In a small app it might be in the `didFinishLaunchingWithOptions` method of the app delegate. Or it might be created “lazily” when it is first needed.

```
self.viewController = [[[ViewController alloc]
initWithNibName:@"ViewController_iPhone" bundle:nil] autorelease];
```

Naming the nib file (“`ViewController_iPhone.xib`”) tells the OS to read that file and initialize everything defined there. The `initWithNibName` method might have very little code in it, only calling the superclass `initWithNibName`.

If you do not use Interface Builder there will be no nib file, and the init message would look like,

```
self.viewController = [[[ViewController alloc] initWithNibName:nil bundle:nil]
autorelease];
```

The OS simply calls the method, assuming that all further initialization will take place in code in `initWithNibName` or `viewDidLoad`.