

Integrating Verious Weather Components in Your Application for iOS

Version 1.0

March 5, 2012

This tutorial is for application developers who will use the Verious Weather component in an application. The following are some steps on how to integrate the Weather framework and how to set up the license manager.

Table of Content

Integrating Verious Weather Components in Your Application for iOS.....	1
Verious Weather Component.....	1
Features:.....	2
Configuration options include:.....	2
Permissions & User Data:.....	2
Integration Steps.....	2
What you need.....	2
Your app.....	3
Add security framework.....	5
Add Weather component.....	7
Include resources.....	8
Add Dependent frameworks.....	8
Add License Manager.....	9
Add UI.....	10
Add an Outlet.....	10
Integrate License Manager.....	10
Implement Code.....	11
Some common bugs.....	13

Verious Weather Component

This powerful component is like a mini app with all of the functionality that you would expect from a standalone weather application. It is the quickest way to add weather data to your Android app, saving you months of development time.

The WDT component would be a perfect addition to applications in categories like; news, sports, travel, city guides, exercise, outdoor activities, and many more.

This component is available in two versions:

Developer - Free with advertisements. This version has the weather API built in along with a popular mobile advertising solution. The revenue from the ads support the cost of the premium weather data.

Enterprise - Paid version where you can run your own ads or keep it free from advertisements. You can try it out with a 30-day free trial, but will have to contact us for pricing. Pricing depends on volume.

Features:

1. Easily-embeddable UI widget – this bar-shaped UI widget can be easily integrated into iOS apps for iPhone and iPod Touch.
2. Location awareness – the weather component queries the device location to show current weather for the current location
3. Ability to Change Location – you can pass the weather component a different location and the user is able to change the location
4. Full Screen 5-day Forecast – displays the daily highs and lows, “feels-like” temp, wind speed and direction.
5. Temperature shown in Centigrade or Fahrenheit – User can toggle between the two

This sophisticated component handles all of the real-time calls, back-end content parsing, data presentation and caching necessary to integrate real-time weather data feeds into a native mobile app.

This component also has comprehensive configuration options so you can change the look and feel to best match your application's user interface.

Configuration options include:

1. Background color
2. Set default centigrade or Fahrenheit
3. Set default location or let it be location aware – Location specific apps like city guides or newspaper apps may want to have it default to a particular city

Permissions & User Data:

1. The component needs access to a data connection and location.
2. Ad supported version also accesses UUID
3. This component does not access any other user data.

Integration Steps

What you need

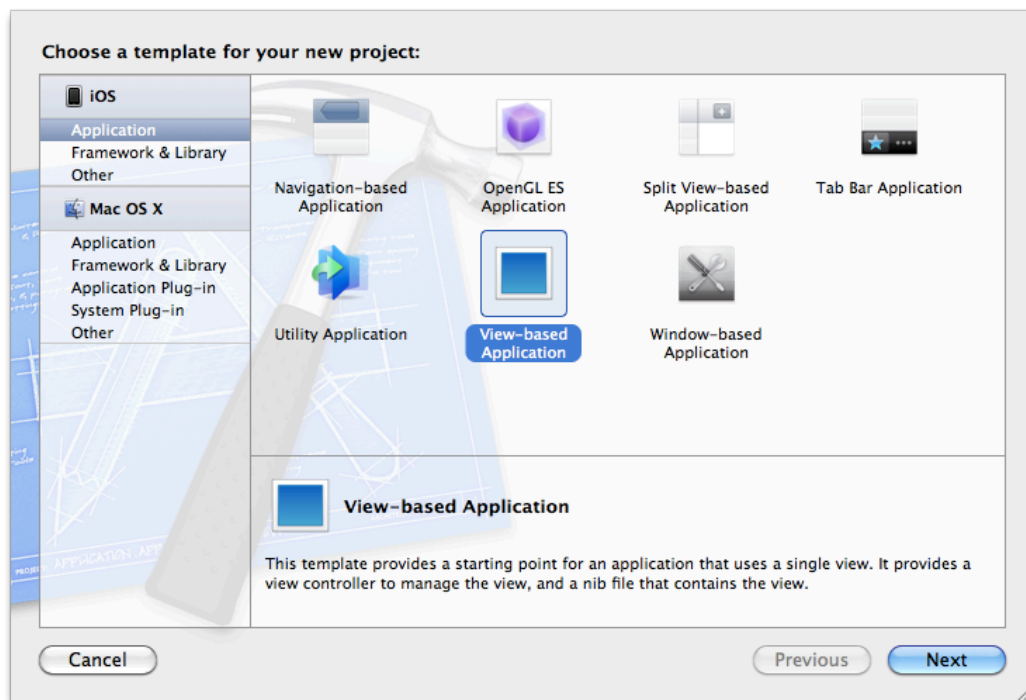
You can download the Verious Weather component from the Verious web site <http://www.verious.com/>. You can search for the component and download(/purchase) it from there.

You will need the Weather framework, the license manager library and security key and token.

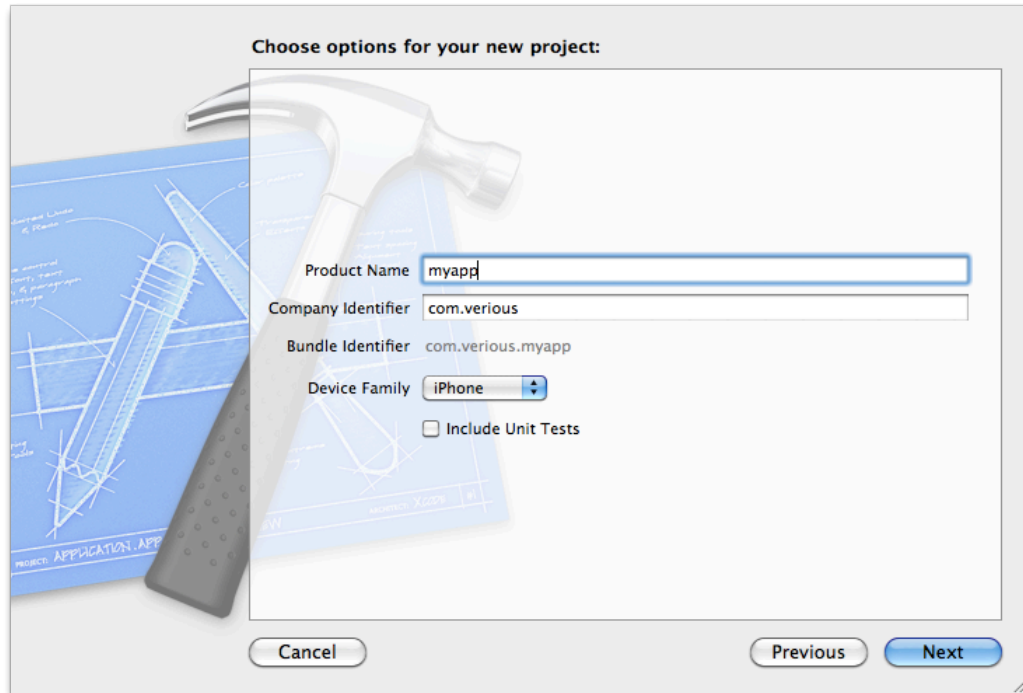
Your app

A test app is already included in the package. You may refer to the test app for implementation details.

You may use your existing app or create a new app first for your integrated app. The following shows you how to create a new app:

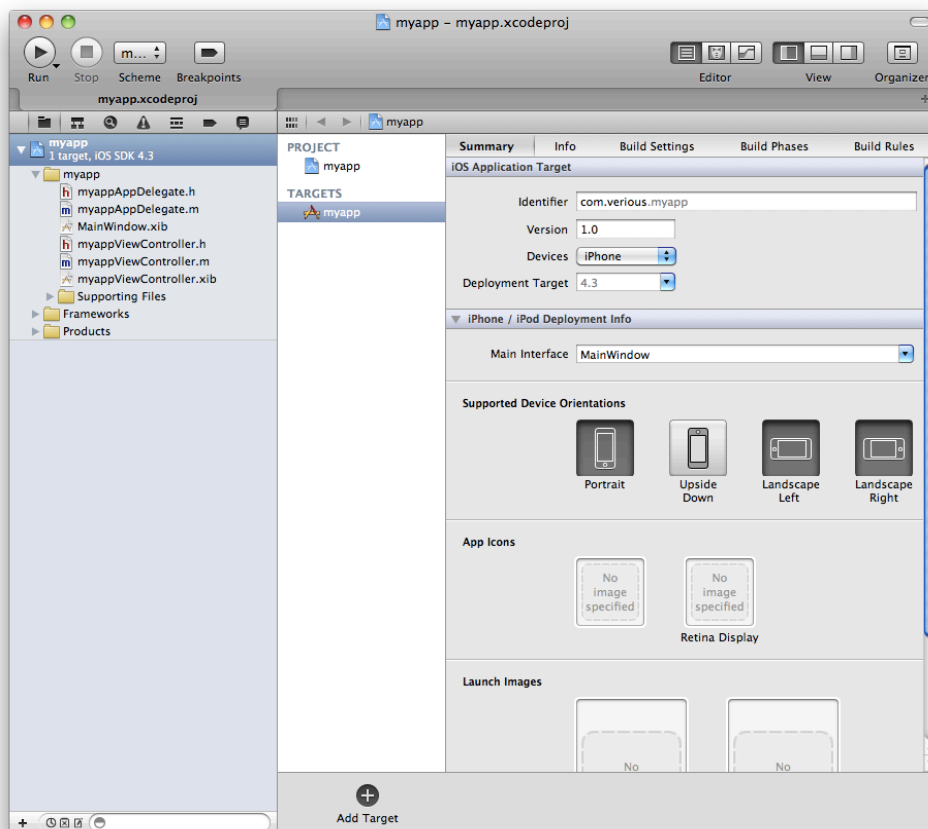


In the menu, select File->New->New Project. On the left panel under the "iOS" section, select "Application". You'll see an icon for developing a "View-based Application". Select the icon and press the "Next" button.



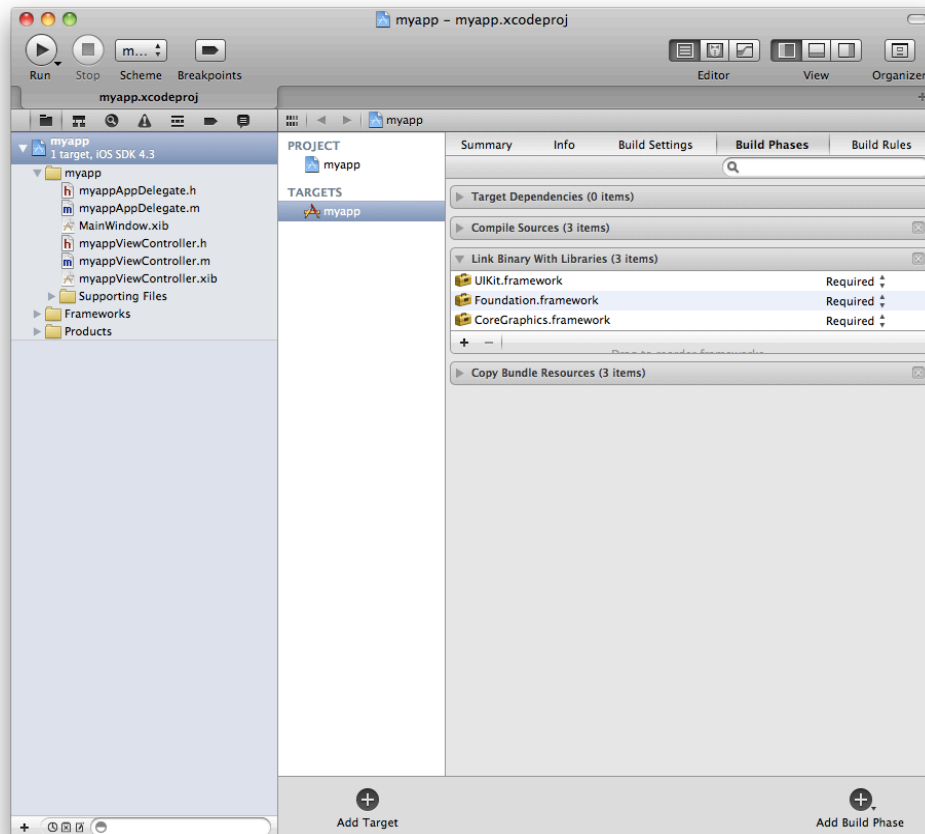
Enter the name for this new application. In this example, we've named the project "myapp"

The company identifier will be "com.verious" as the license key is locked to the iOS app bundle identifier "com.verious.myapp". Press the "Next" button and specify the path to store your project. The result should appear as below:

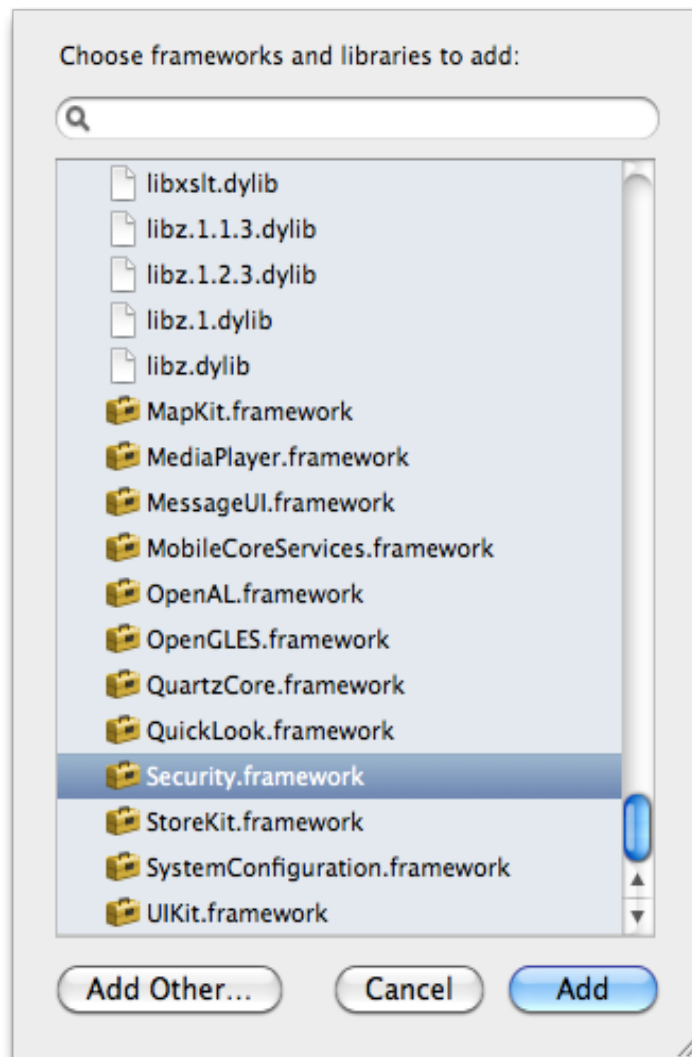


Add security framework

The license manager integration will need the security framework. The following shows how to add the framework into your app. The example is shown with myapp, but it should be the same with your own app.



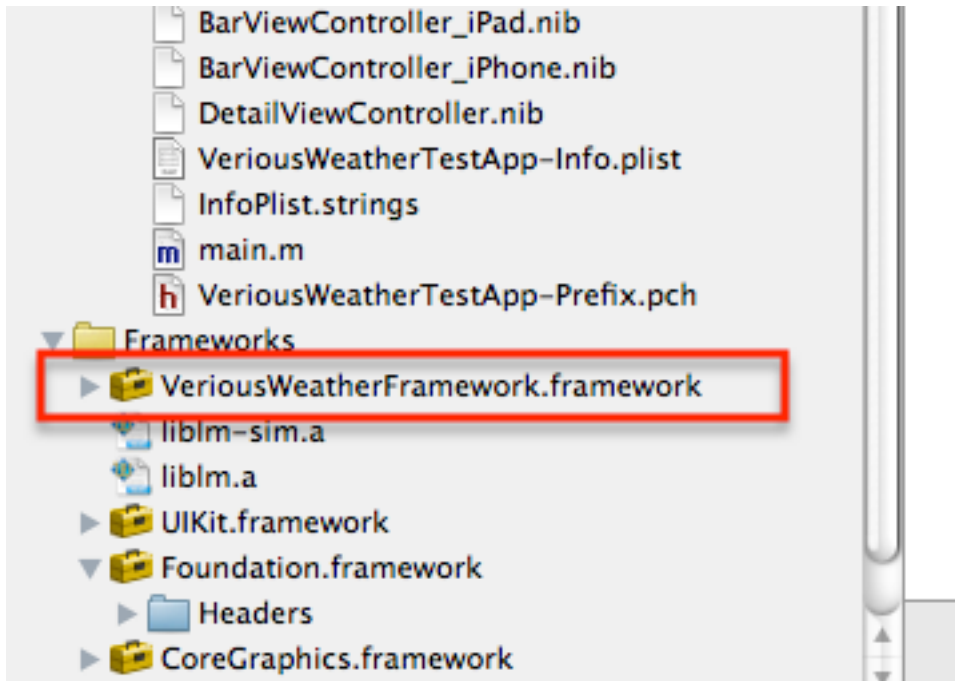
From the "myapp" project details tree node, select the "Build Phases" tab. Expand the "Link Binary With Libraries" section. Then press the "+" button to add a new framework.



Select the "Security.framework" to add to your project and press the "Add" button.

Add Weather component

You can drag and drop the framework VeriousWeatherFramework.framework into your project



Include resources

The weather framework has a resource folder that contains images and nib files. You need to add these files to your project.

Add Dependent frameworks

The weather framework depends on some other framework. You need to add these frameworks like how you added the security framework earlier.

The dependent frameworks include:

- QuartzCore
- AVFoundation
- AudioToolbox
- SystemConfiguration
- MobileCoreServices
- MessageUI
- CoreLocation
- CFNetwork
- MediaPlayer
- Foundation
- UIKit
- CoreGraphics

It also depends on some dynamic libraries like:

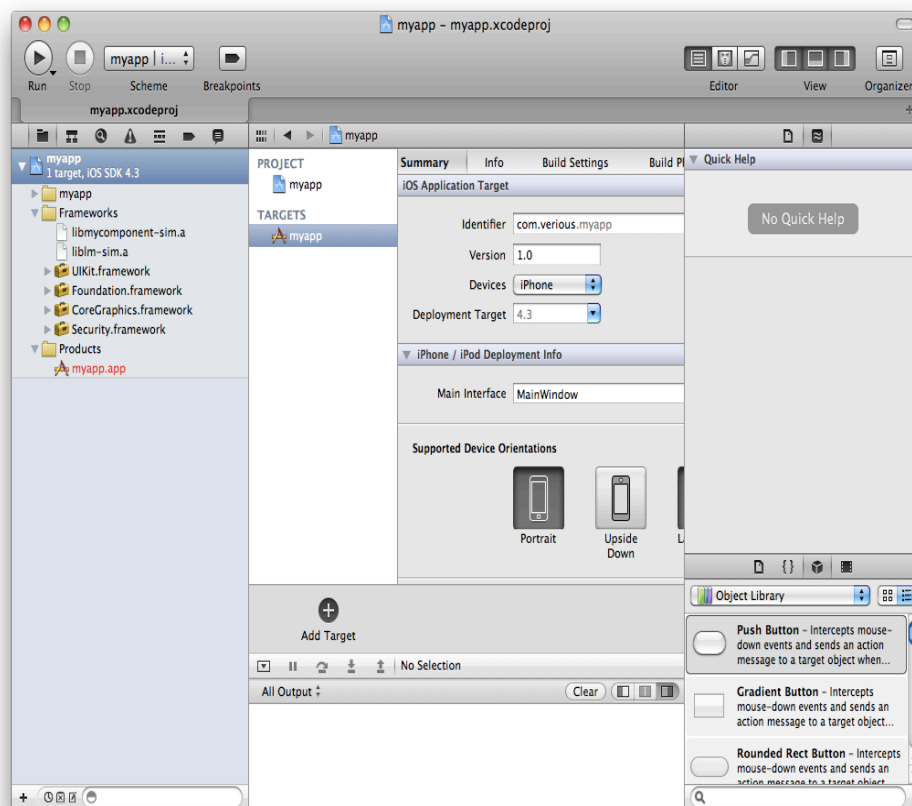
Libxml2.kylib

Libz.dylib

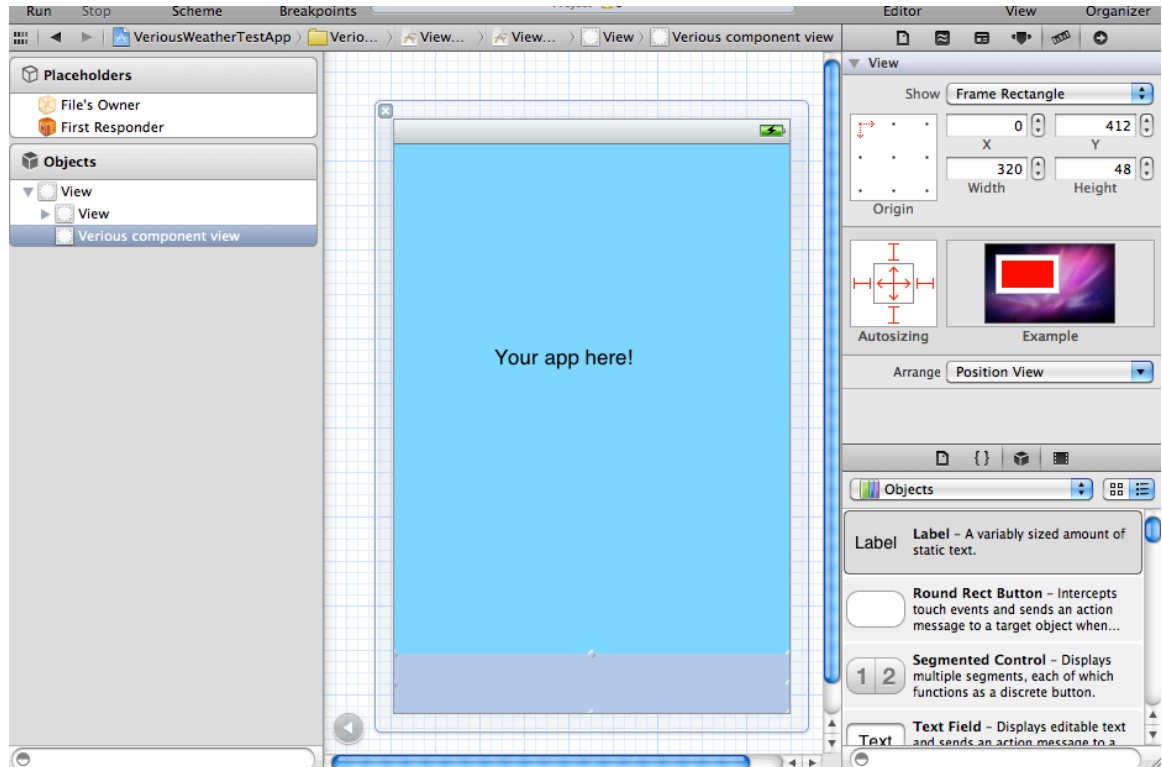
You can add these libraries the same way.

Add License Manager

Add the license manager and component files including: "liblm-dev.a", "liblm-sim.a", and "LicenseMgr.h".



Add UI



Add a view for the Verious component. This view should have the dimensions as seen here.

Add an Outlet

Create an outlet for the Verious Component View.

Integrate License Manager

First, let's edit the view controller file (in the test app is called "ViewController.h"). Add a new property called "status". And while we're at it, we'll implement the LMCompletionDelegate from the license manager. This delegate is invoked when the license response is received from the Verious license validation server. It lets the application developer know that it can start calling component functions.

```
#import <UIKit/UIKit.h>
#import <VeriousWeatherFramework/BarViewController.h>
#import <VeriousWeatherFramework/Weather.h>
#import "LicenseMgr.h"

@interface ViewController : UIViewController
<LMCompletionDelegate> {
    BarViewController *barViewController;
```

```
}
```

```
@property (retain, nonatomic) IBOutlet UIView *testView;  
@property (retain, nonatomic) NSString *status;
```

```
- (void)licenseCheckCompleted;
```

Implement Code

In viewController.m:

1. implement licenseCheckCompleted;

```
- (void)licenseCheckCompleted {  
    NSLog(@"Done checking licenses");  
}
```

In viewDidLoad, make the license manager register and validation:

```
[[LicenseMgr instance]  
    registerLicense:@"1d42d0f0009b60d76edc01d844d5a657"  
    component:barViewController];  
[[LicenseMgr instance] validateLicenses:self];
```

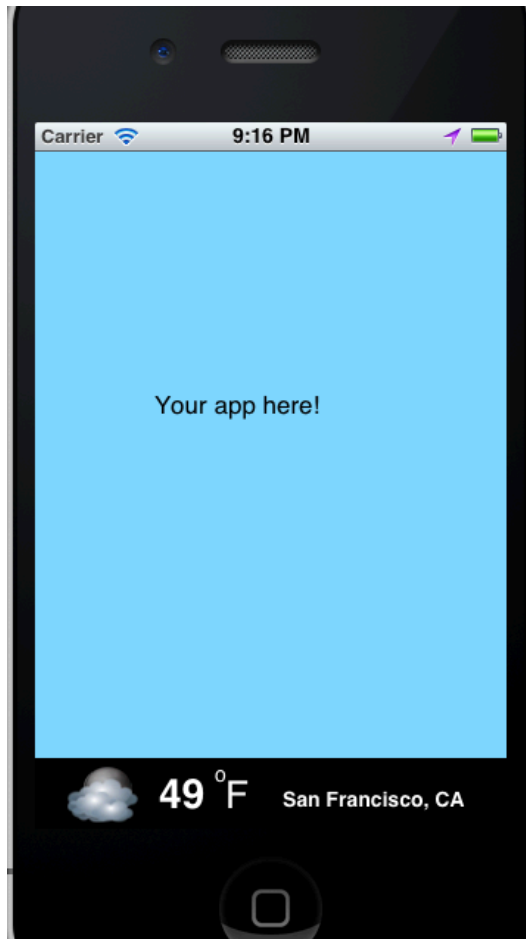
Copy the security token from your downloaded token file to replace the string above in the register call.

To connect the view to the weather component view BarViewController, call:

```
barViewController = [[BarViewController alloc]  
    initWithNibName:fileName bundle:nil];  
[testView addSubview:barViewController.view];
```

1. Build and Run Your Application

Now, build and run your application.



and when you click the bar, you should see the full view:



Some common bugs

You may get this bug:

BarViewController.h: Semantic Issue: Unknown type name 'class'

This is because of a C++ code dependency in BarViewController.h. You may need to change AppDelegate.m and some other files to .mm. Simply click on the file name, and change .m suffix to .mm.