

从零开始学iOS7开发系列3-我的地盘我做主-Cha6

原文及示例代码来自raywenderlich store中的iOS Apprentice 系列3教程，经过翻译和改编。

版权归原作者所有，本系列教程仅供学习参考使用，感兴趣的朋友建议购买原英文教程教程(The iOS Apprentice Second Edition: Learn iPhone and iPad Programming via Tutorials!)

在火炉边找个位子坐吧，终于重复的快没词了，看来暴雪这个细节还是有待加强啊。欢迎继续我们的学习。

好消息，从这一课开始，我们会进入真正的移动应用开发了。因为我们将介绍如何使用Core Location框架来获取用户的经度 (longitude)和纬度(latitude)信息。

额，这个经度和纬度是啥意思？地理盲表示不懂啊。

因为这里不是地理课，就不长篇大论来解释这两个名词了。作为开发者，最重要的是知道，通过经度和纬度信息，就可以精确定位一个人所在的位置。如果你想了解更多细节，可以参考：<http://baike.baidu.com/subview/93982/12002011.htm>

还记得很久之前那部贺岁片《手机》的结尾吗？是的，的确是太久了，这部电影的上映时间是2004年，都10年过去了！那时候的手机能够录制欢乐时光的视频就已经让人非常羡慕了，更让人惊叹不已的是，在影片的结尾处，竟然可以使用手机来定位人的具体位置。当时我伙呆，如今想来，冯导还真是个有极客精神的the one啊！

如今智能手机没有GPS定位功能只有一个可能-山寨仿冒！不不不，即便是山寨仿冒的300元一个的android手机也都把这个当做标配功能了。

去年（2013年）看到大量可穿戴设备的出现，在各种五花八门的智能手环中，我个人最感兴趣的还是360的儿童手环了。虽然我对360和企鹅都没太大的兴趣，但可以跟踪孩子所在位置的儿童手环还是让我非常心动，毕竟我也是家长级别了~

What's the next big thing?是Google的Glass？还是苹果的iWatch?还是iTV? 让我们共同期待精彩的2014年吧。



言归正传，在本节内容完成后，我们会看到类似下面的界面：

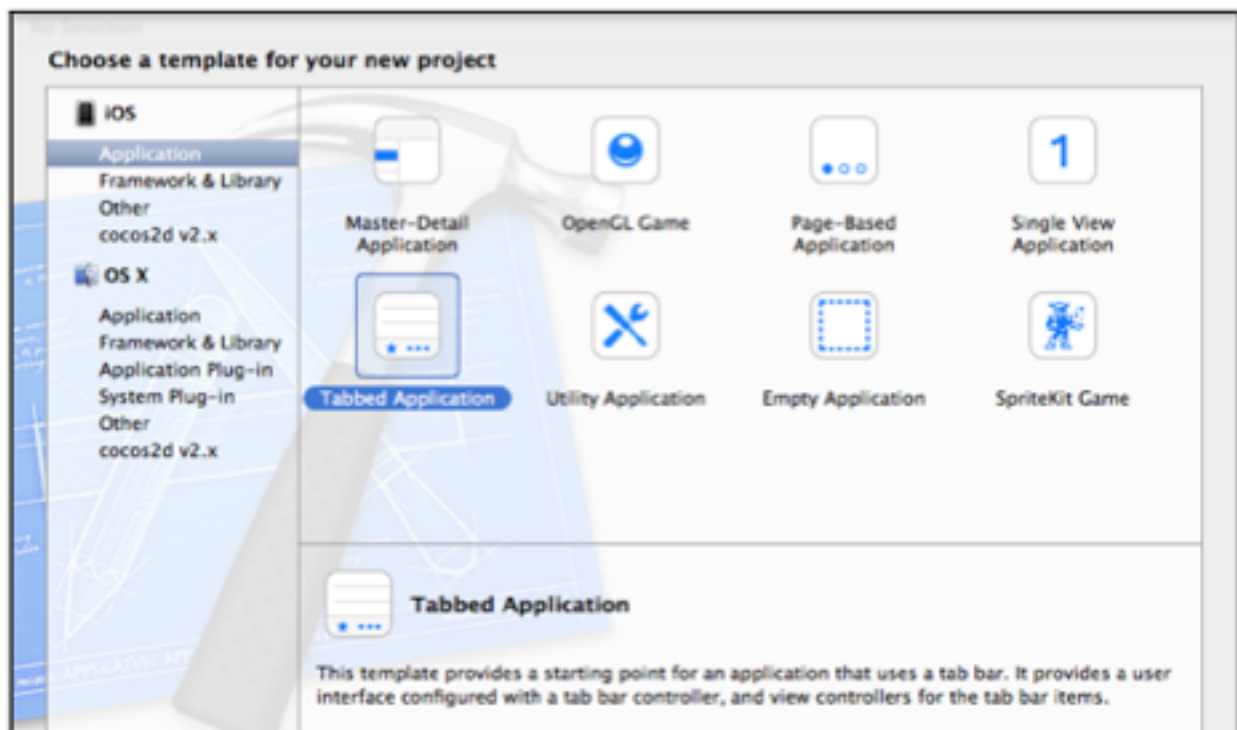


The first screen of the app

我得承认这个界面实在是够烂的，不过不要紧，后面我们会把它完善。毕竟当前的第一要务是让你尽快学习如何获取手机用户的GPS坐标信息，然后显示在屏幕上。这就是互联网思维的产品开发模式，先完成，再完善。当然，不要以偏概全，不是说让你提交给最终用户的时候也是这样，而是说开发的过程中应该如此。

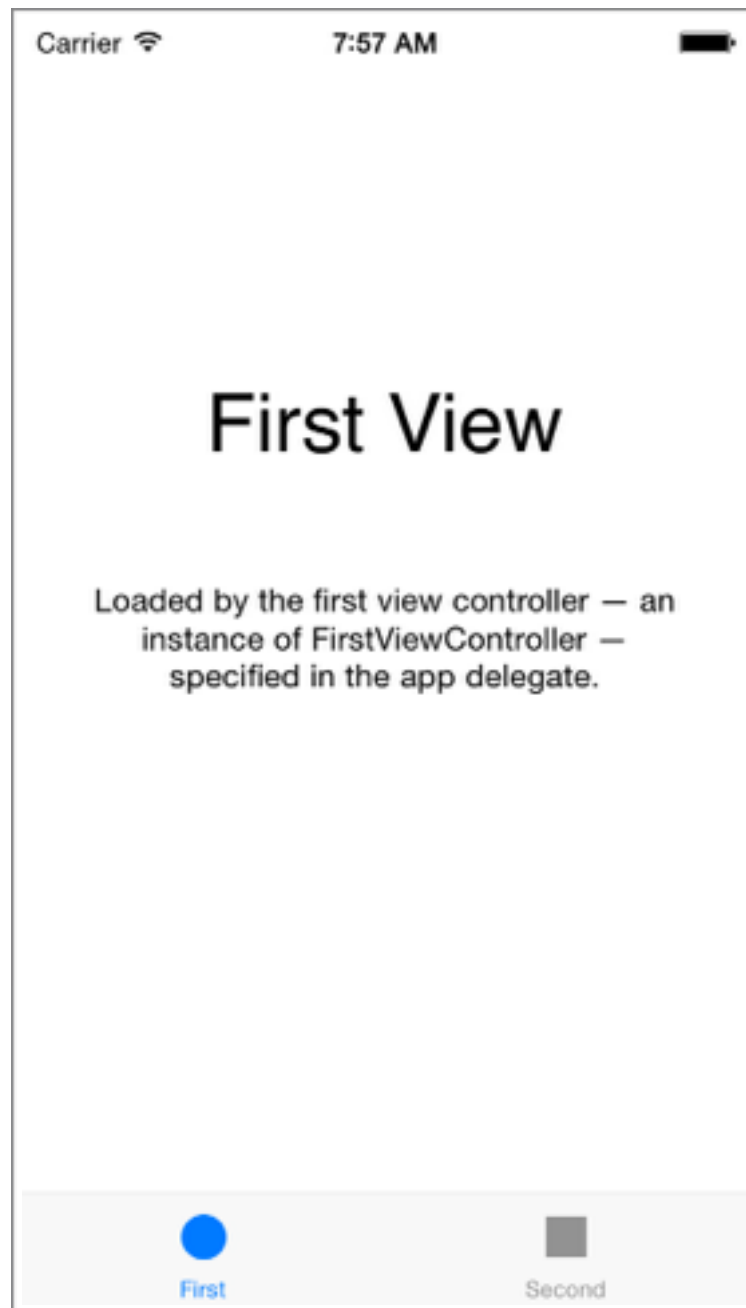
打开Xcode（久违了！），选择Tabbed Application这个模板：

然后在下面的界面中输入以下信息：



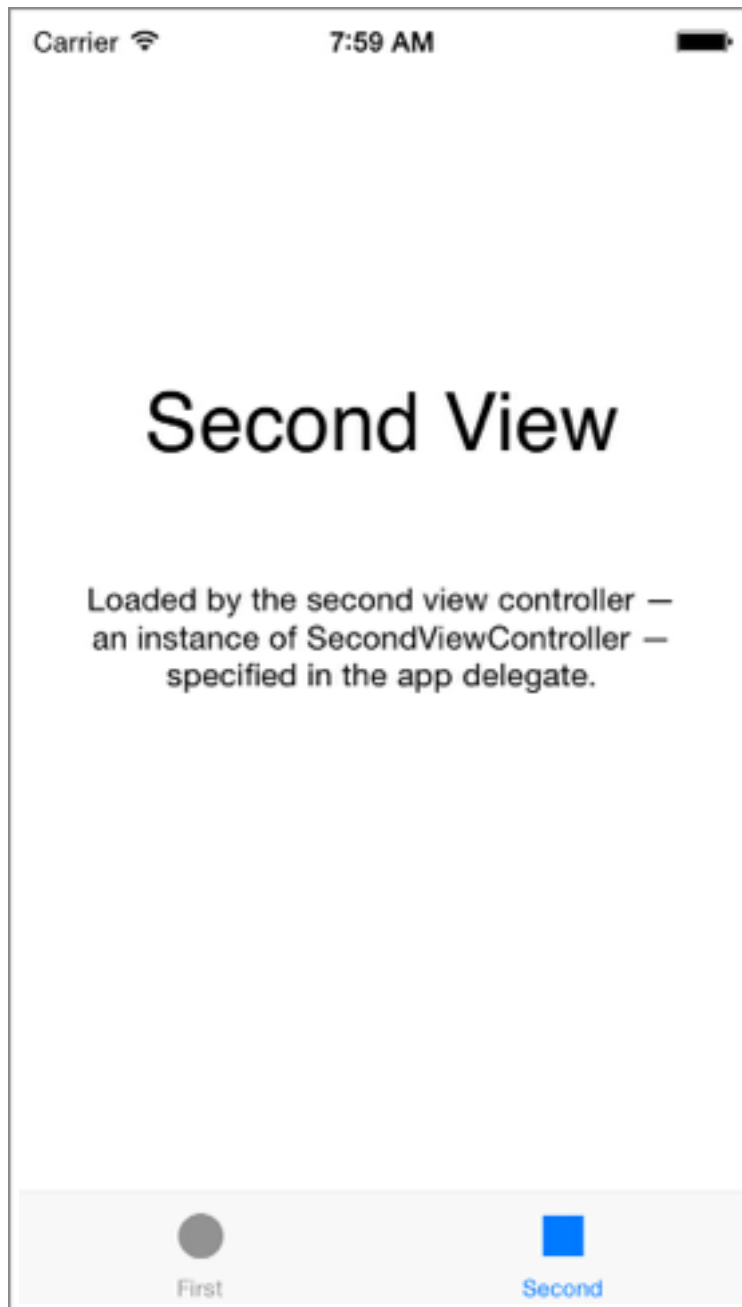
- 1.Product Name: MyLocations
- 2.Organization Name:你或者你公司的名称（英文）
- 3.Company Identifier: 你的唯一标识符（通常是反转域名的表示法，也就是com.myIdentifier）
- 4.Class Prefix:类的前缀，这里可以留空白
- 5.Devices:选择iPhone
- 6.点击next保存项目。

编译运行项目，会看到类似下面的界面：



在应用的底部有一个tab bar，其中包含了两个tab:First和Second

你可以试着在两个Tab间来回切换。



虽然这个应用的功能也就到此为止了，不过它已经有了三个视图控制器：作为root controller的UITabBarController，它的作用就是容纳tab bar（标签栏，或者有人翻译成分页栏，随你喜欢，我更喜欢直接用英文原文）。使用UITabBarController可以在不同的界面之间来回切换。

小提示：

目前来说大量的iOS应用都采用tab bar或者侧滑菜单的设计方式，在知乎上有关于此的讨论，侧滑菜单（left nav）可以让主视图的显示区域更大，但切换视图需要用户分两步来完成（拉开菜单，点击切换），而tab bar的优点是切换方便，但缺点是占用部分的屏幕显示面积。

如果你是个设计人员，可以参考这里：

<http://jxnblk.tumblr.com/post/36218805036/hamburgers-basements-why-not-to-use-left-nav-flyouts>

关于侧滑菜单的实现在cocoachina和code4app中有很多示例demo可供参考：

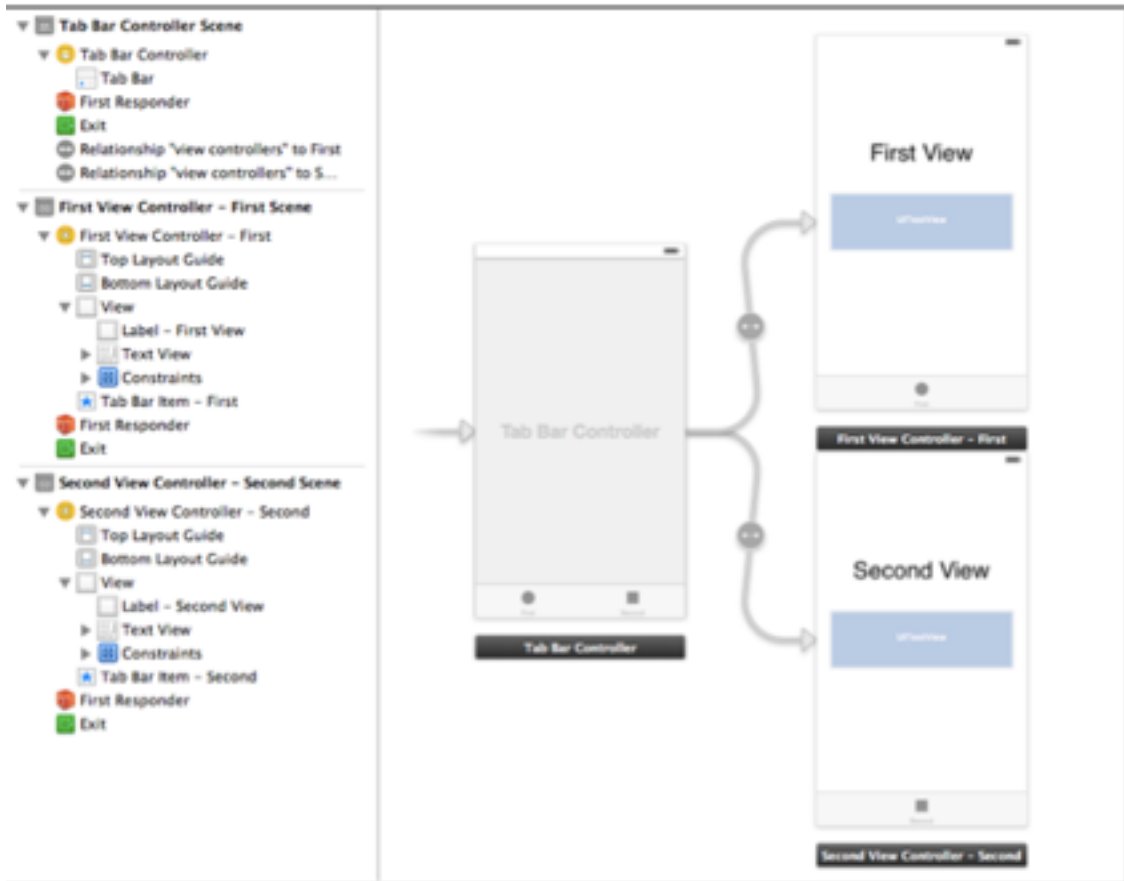
http://code.cocoachina.com/index/0/?seach_flg=1&key_word=侧滑

<http://code4app.com/search/侧滑菜单>

提示完毕。

继续刚才的内容。在这个模板创建的应用中除了UITabBarController，还有分属于两个tab（标签栏）的视图控制器，Xcode默认将它们的名称设置为FirstViewController和SecondViewController

此时的storyboard会是类似下面的样子：



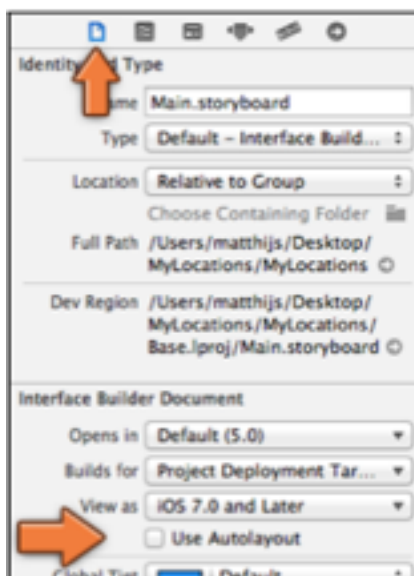
The storyboard from the Tabbed Application template

好吧，在我可怜的13寸Macbook Air上面显示storyboard的全貌实在是太费功夫了。除了缩放，别无他法，storyboard虽好，也要好屏幕相配啊！如果你买得起27寸的iMac，或者15寸的Macbook Pro，都是不错的选择啊。

在这部分内容中，我们暂时只会用到第一个tab选项。当然后面我们不光会用到第二个tab选项，还会添加第三个tab。

现在我们还有一件事情要做，就是禁用Auto Layout。虽然Auto Layout是创建复杂用户界面的好方法，不过对目前这个应用来说就有点大材小用了。并非任何时候都适合用Auto Layout，只在需要的时候才使用这个技术。

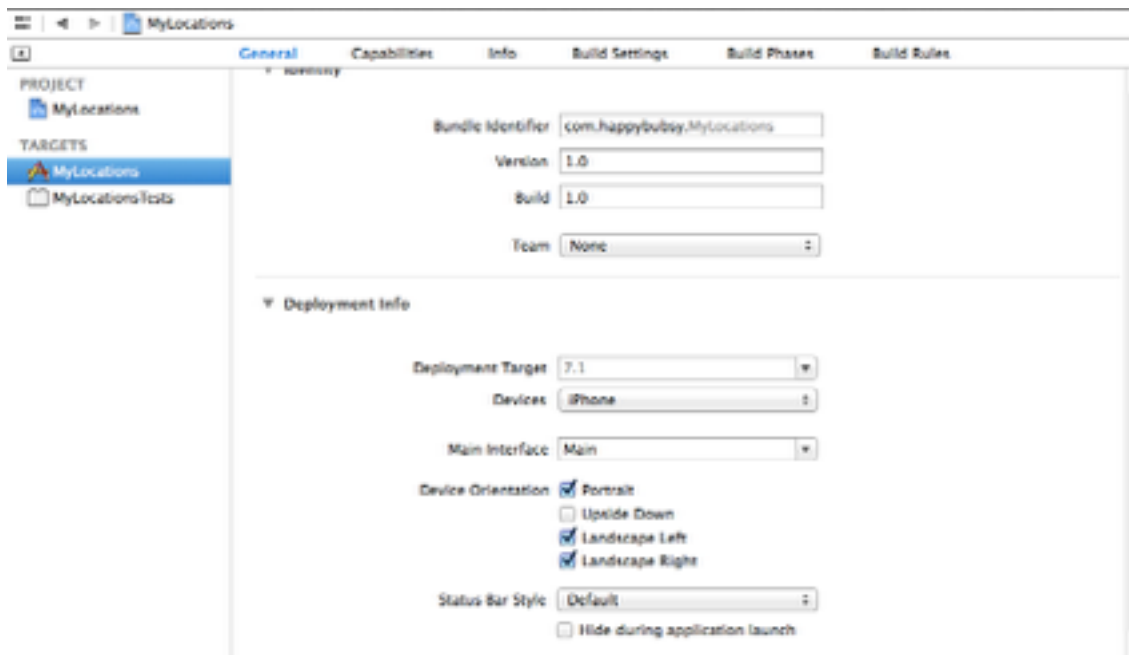
在Xcode中切换到storyboard，在右侧面板中切换到File inspector，然后取消对Use AutoLayout选项的勾选。



首先让我们给FirstViewController取个好名字，在Xcode中切换到FirstViewCotnroller.h，把鼠标光标放到@interface后面的FirstViewController上面，然后从Xcode的顶部菜单中选择Edit-Refactor-Rename，然后在对话框中输入CurrentLocationViewController。具体的操作就不重复了，如果你自己还搞不定，可以回过头看看上一系列的教程中是怎么操作的~就当是练习了。

注意要确认Xcode在storyboard中也更改了视图控制器的Class属性（和第一个tab相关联的）。我们可以在Identity inspector中看到这一信息。Xcode应该已经自动帮我们完成了这项工作，不过万一没有呢？

接下来在Xcode中切换到项目设置界面，然后在Deployment Info部分的Device Orientation子部分取消勾选Landscape Left和Landscape Right，只保留Portrait。

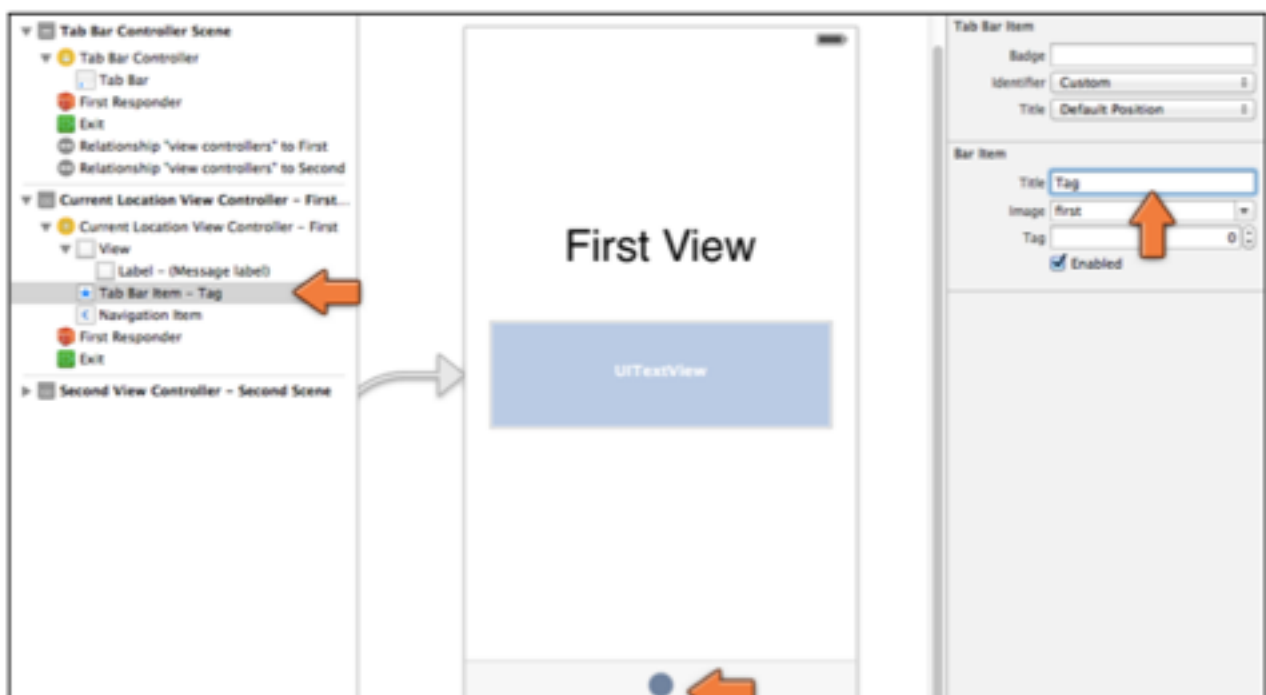


编译运行应用，确保一切工作正常。

我有个小小的习惯，每当更改了任何和storyboard有关的东西后，都会第一时间重新编译运行应用。人多忘事，有时候一个小小的步骤遗漏都会导致应用无法正常工作。

在之前学习navigation controller（导航控制器）的时候，我们了解到，内置在导航控制器内的控制器在顶部会有一个Navigation Item对象，可以用它来配置导航栏。Tab bar（标签栏）的工作原理是类似的，代表一个tab的每个视图控制器都有自己的Tab Bar Item对象。

选中Current Location View Controller的Tab Bar Item对象，然后在Xcode右侧的面板中切换到Attributes inspector。把Title更改为Tag。后续我们还会设置它的图像，不过默认是一个小圆形。



好了，现在我们可以来设置第一个tab选项卡的界面了。这个界面很简单，只有两个按钮，以及几个用来显示GPS坐标和地址的标签。为了节省时间，让我们一次添加所有的outlet属性。

在Xcode中切换到CurrentLocationViewController.h，然后添加代码如下：

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

@interface CurrentLocationViewController : UIViewController

@property(n nonatomic,weak) IBOutlet UILabel *messageLabel;
@property(n nonatomic,weak) IBOutlet UILabel *latitudeLabel;
@property(n nonatomic,weak) IBOutlet UILabel *longitudeLabel;
@property(n nonatomic,weak) IBOutlet UILabel *addressLabel;
@property(n nonatomic,weak) IBOutlet UIButton *tagButton;
@property(n nonatomic,weak) IBOutlet UIButton *getButton;

-(IBAction)getLocation:(id)sender;

@end
```

好久没写代码了吧？为了保持手感，建议所有的代码都一行行手动敲入。如果你想copy paste，随你，不过懒人的下场就是-很快你就会遇到更多更复杂的代码，然后你看了就烦，然后就没有然后了。。。

日敲代码三百行，不会编程也会编。无它，唯手熟尔。

如果说在写代码前的思考是有意识的行为，那么每天习惯性的敲一些代码则是培养潜意识中对代码的感觉。即便是高手，如果长时间不写代码，也会觉得生疏，更何况是我们这种菜鸟。

如果说只写代码不思考就会沦为码农，那么只思考不写代码最终只会让自己最终失去编程的乐趣。

好了，接下来在storyboard中设计UI如下：



其中：

1.顶部的message label横跨整个屏幕。我们会使用这个标签来显示应用在获取GPS坐标时的状态信息。将Alignment属性设置为center，然后把它和刚才的messageLabel关联起来。

神马？忘了怎么操作？哼哼，这里我可不会那么客气了。自己去翻翻上一篇教程吧。

2.将(Latitude goes here)和(Longitude goes here)这两个标签设置为右对齐，然后分别关联到latitudeLabel和longitudeLabel上。

3.(Address goes here)标签同样横跨整个屏幕，它的高度为50points，这样就可以容纳两行文字了。将Lines属性设置为0（意味着它可以自适应可变的行数）。然后将其关联到addressLabel上。

4.Tag Location按钮目前还不需要做任何事情，不过还是应该将其关联到tagButton这个outlet属性变量上。

5.把Get My Location按钮关联到getButton 这个outlet属性变量上，同时让它的Touch Up Inside event关联到getLocation:这个动作方法上。

以上这些操作，我会告诉你具体怎么做吗？显然不会，哈哈，上一系列教程不好好学的童鞋可以歇歇了。回过头再好好学一遍吧，谁让你偷懒来着~

为了避免我们触碰按钮时应用崩溃，还需要在CurrentLocationViewController.m中添加动作方法的实现代码：

```
-(IBAction)getLocation:(id)sender{
```

```
    //占位的，后面再补充
```

```
}
```

好了，现在来编译运行项目，看看我们的新设计成果。

注意，里面的标签和按钮上的文字可以根据自己的喜好随意更改。



注意：如果你用的是3.5-inch 的Simulator，先切换到iPhone Retina(4-inch)或者iPhone Retina(4-inch 64-bit)。在这个教程中，我们会优先适配4寸的设备。当然后续会把它适配到3.5寸的设备上。

好吧，就这样，我们接触了一个全新的界面设计利器-Tab bar。
等等，说好的GPS定位呢？怎么搞？预知后事如何，且听下回分解。

先送上福利（2014年1月23日）

最近很火的湖南女财神，看看吧，其实我对红包更感兴趣。



有人不满了，此等庸脂俗粉不堪入目，还是看看我的二次元萌妹子吧。

