

iOS学习之Table View的简单使用

分类: [iOS开发入门](#)

2012-06-07 20:17

9804人阅读

[评论\(22\)](#)

[收藏](#) [举报](#)

[header](#) [image](#) [Image](#) [ios](#) [iOS](#) [IOS](#) [table](#) [xcode](#)

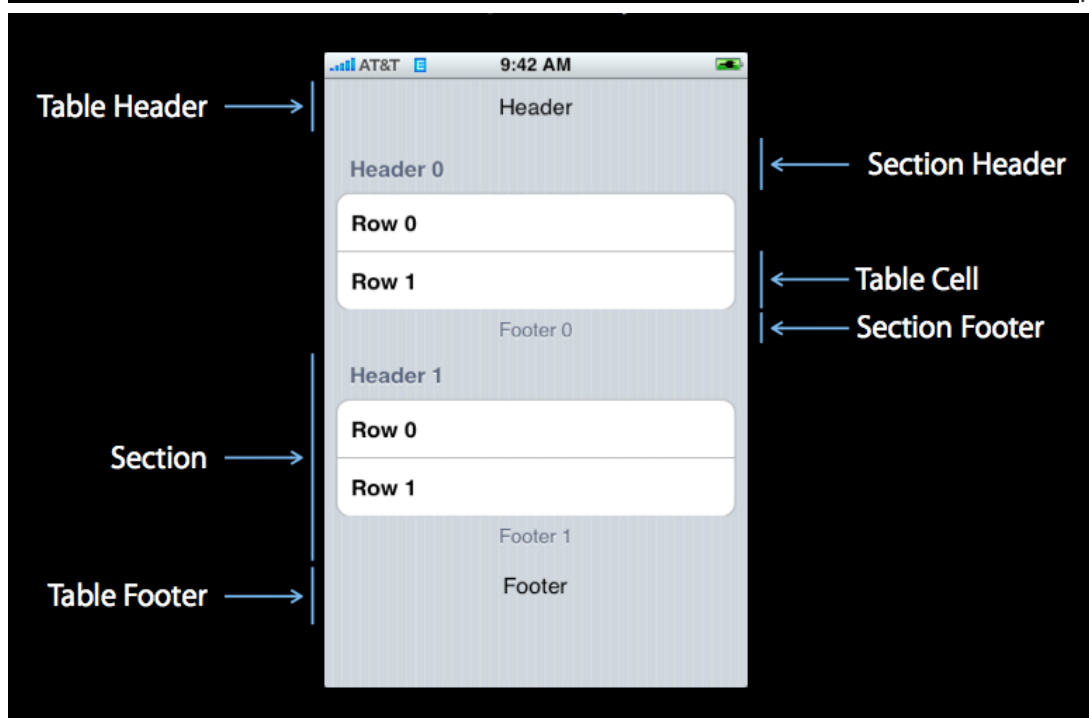
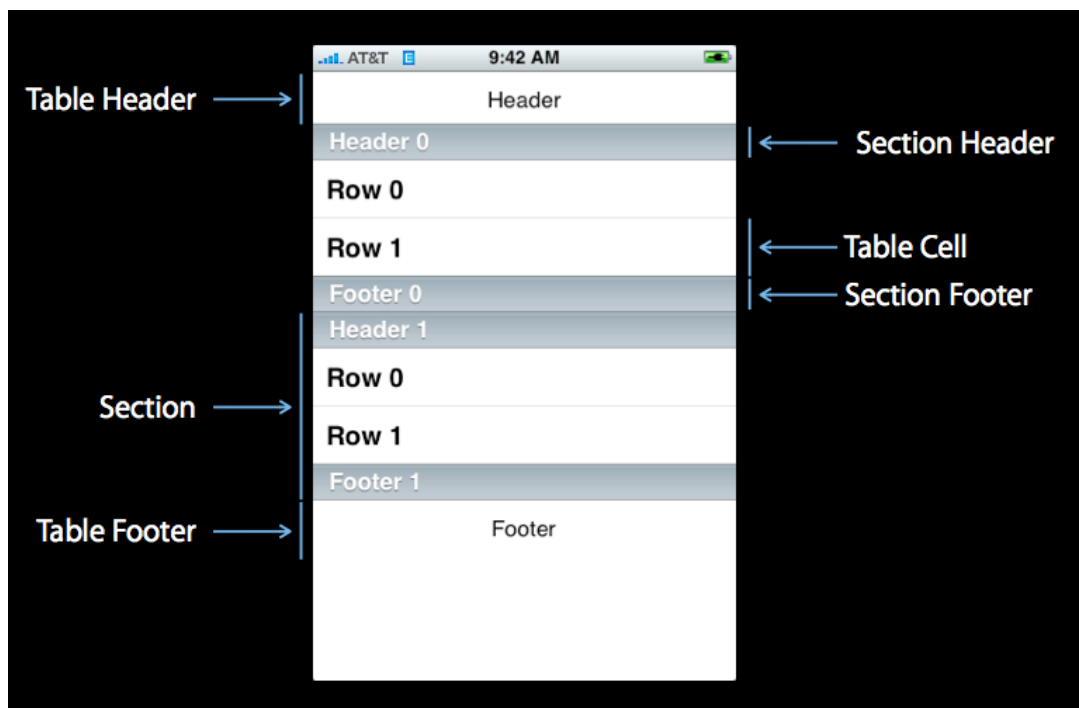
Table View简单描述:

在iPhone和其他iOS的很多程序中都会看到Table View的出现, 除了一般的表格资料展示之外, 设置的属性资料往往也用到Table View, Table View主要分为以下两种:



- Plain: 这是普通的列表风格
- Grouped : 这是分块风格。

对于UITableView, 我們有一些特殊的概念和术语, 比如说我们成Table View的一行为Cell, 而许多的Cell可以组成Section, 每个Section上下又分别有Header和Footer, 许多个的Section则组成了整个Table , 当然Table也有Header和Footer, 下面看两种图就能明白这几个拗口名词了:



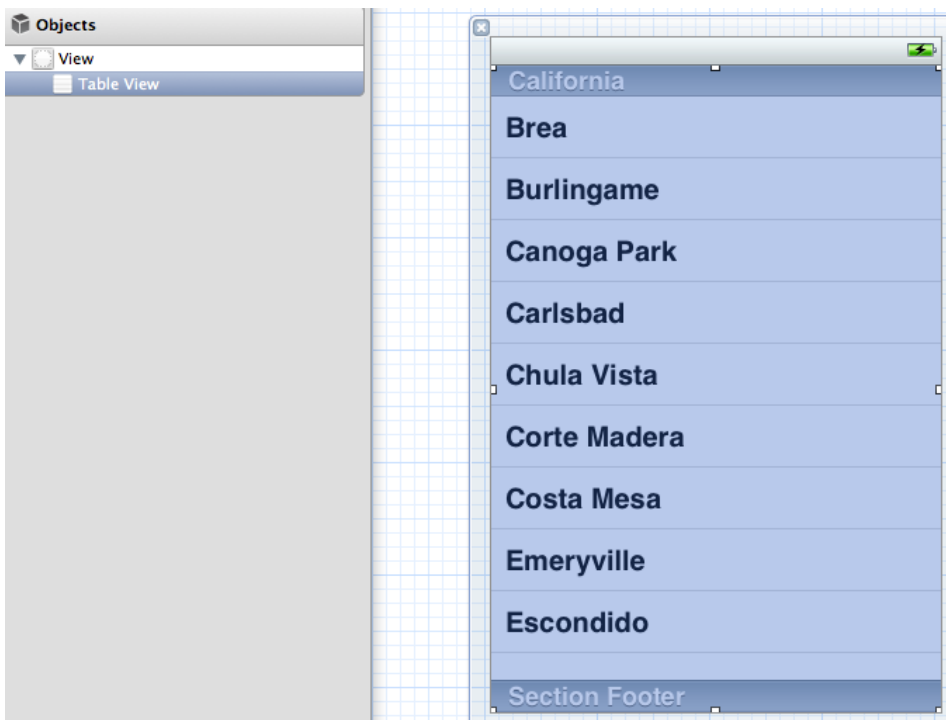
现在理论知识了解的差不多了。今天先做一个Plain样式的例子，这样加强对Table view的熟练使用。

1、新建项目

新建一个Single View Application,命名为TableViewDemo，开发环境是：Xcode 4.3,iPhone 5.1模拟器。

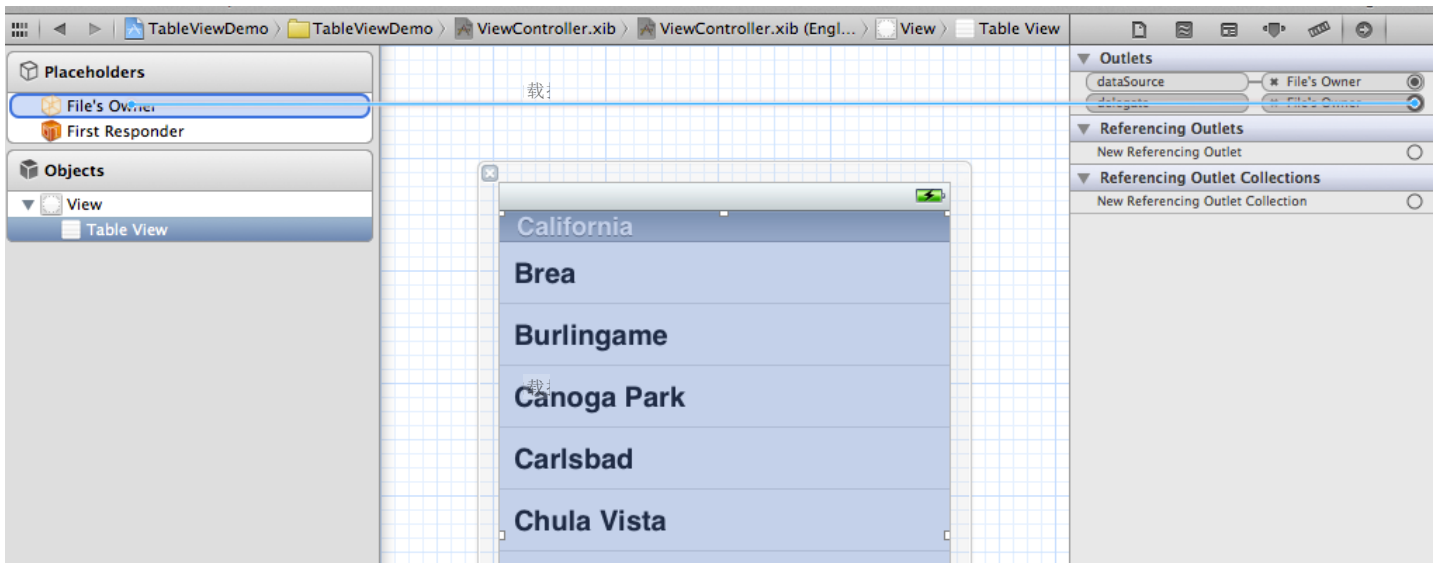
2、Table View放上控件

打开ViewController.xib文件，往ViewController.xib界面上拖拽一个Table View控件到现有的View上，对齐。



3、连接新添加的Table View和ViewController。

选中新添的Table View控件，打开连接检查器(Connection Inspector)，找到delegate和datasource并点中圆圈拉线连接到左边File's Owner图标上，为什么要把这两个连接File's Owner上呢？这是因为iOS使用的MVC设计模式，View和ViewController之间的对应关系，需要一个桥梁来进行连接的（即，对于一个视图，他如何知道自己的界面的操作应该由谁来响应），这个桥梁就是File's Owner。

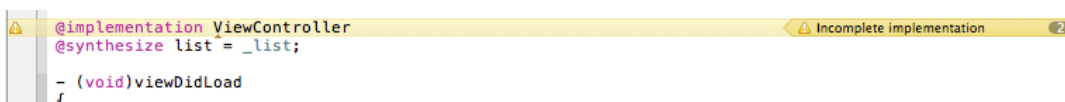


4、打开ViewController.h,添加协议和Property (类似与java里的实现接口)

```
[cpp]
01. #import <UIKit/UIKit.h>
02.
03. @interface ViewController : UIViewController<UITableViewDelegate, UITableViewDataSource>
04. @property (strong, nonatomic) NSArray *list;
05. @end
```

5、打开.m文件，添加：

```
[cpp]
01. @synthesize list = _list;
```



这是发现有两个警告，提示未完成的实现，这提示的是UITableViewDelegate, UITableViewDataSource这个两个头文件里的协议的方法未实现。待会我们去实现它。

6、建立数据

```
[cpp]
```

```

01. - (void)viewDidLoad
02. {
03.     [super viewDidLoad];           载!
04.     // Do any additional setup after loading the view, typically from a nib.
05.     NSArray *array = [[NSArray alloc] initWithObjects:@"美国", @"菲律宾",
06.                 @"黄岩岛", @"中国", @"泰国", @"越南", @"老挝",
07.                 @"日本", nil];
08.     self.list = array;
09. }
10.
11. - (void)viewDidUnload
12. {
13.     [super viewDidUnload];
14.     // Release any retained subviews of the main view.
15.     self.list = nil;
16.
17. }

```

7、生成row

关键的步骤来了，实现tableView添加数据源，返回TableView的行数，返回各行cell实例。

```

[cpp]
01. - (UITableViewCell *)tableView:(UITableView *)tableView
02.     cellForRowAtIndexPath:(NSIndexPath *)indexPath {
03.
04.         static NSString *TableSampleIdentifier = @"TableSampleIdentifier";
05.
06.         UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:
07.             TableSampleIdentifier];
08.         if (cell == nil) {
09.             cell = [[UITableViewCell alloc]
10.                 initWithStyle:UITableViewCellStyleDefault
11.                 reuseIdentifier:TableSampleIdentifier];
12.         }
13.
14.         NSUInteger row = [indexPath row];
15.         cell.textLabel.text = [self.list objectAtIndex:row];
16.         return cell;
17.     }

```

上面的第二个方法中，

`UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier: TableSampleIdentifier];`

这个语句根据标识符TableSampleIdentifier寻找当前可以重用的UITableViewCell。当某行滑出当前可见区域后，我们重用它所对应的UITableViewCell对象，那么就可以节省内存和资源。

这里UITableViewCellStyleDefault是表示UITableViewCell风格的常数，除此之外，还有其他风格，后面将会用到。

注意参数(NSIndexPath *)indexPath，它将行号row和部分号section合并了，通过[indexPath row];获取行号。之后给cell设置其文本：

`cell.textLabel.text = [self.list objectAtIndex: row];`

8、现在一个简单的TableView就弄好看，运行下看效果



9、添加图片。

在项目上add files到项目，提交两张小图片，然后在cell返回之前添加如下代码

```
[cpp]
01. NSInteger row = [indexPath row];
02.    cell.textLabel.text = [self.list objectAtIndex:row];
03.    UIImage *image = [UIImage imageNamed:@"qq"];
04.    cell.imageView.image = image;
05.    UIImage *highLighedImage = [UIImage imageNamed:@"youdao"];
06.    cell.imageView.highlightedImage = highLighedImage;
07.    return cell;
```

效果如下：



载:

10、设置行的风格

表示UITableViewCell风格的常量有:

UITableViewCellStyleDefault

UITableViewCellStyleSubtle

UITableViewCellStyleValue1

UITableViewCellStyleValue2

可以自己修改看看效果。可以添加一个detail

```
cell.detailTextLabel.text=@"打打打打";
```

```
return cell;
```



11、选择table里的某一行

在.m文件@end之前编写 -(void)table 这时会自动提示可以实现的方法，

```

17 (cell == nil) {
    cell = [UITableViewCell alloc]
    tableView:(UITableView *)tableView accessoryButtonTappedForRowWithIndexPath:(NSIndexPath *)indexPath
    tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)editingStyle for...
    tableView:(UITableView *)tableView didDeselectRowAtIndexPath:(NSIndexPath *)indexPath
    tableView:(UITableView *)tableView didEndEditingRowAtIndexPath:(NSIndexPath *)indexPath
    tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath
    tableView:(UITableView *)tableView moveRowAtIndexPath:(NSIndexPath *)sourceIndexPath toIndexPath:(N...
    tableView:(UITableView *)tableView performAction:(SEL)action forRowAtIndexPath:(NSIndexPath *)index...
    tableView:(UITableView *)tableView willBeginEditingRowAtIndexPath:(NSIndexPath *)indexPath
    tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSInd...
}
-(void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath

```

我们选择这个方法

`-(void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath`

选中是做提示，提示选中了那个信息，代码实现如下：

```

[cpp]
01. -(void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath{
02.     NSString *rowString = [self.list objectAtIndex:[indexPath row]];
03.     UIAlertView * alter = [[UIAlertView alloc] initWithTitle:@"选中的行信
    息" message:rowString delegate:self cancelButtonTitle:@"确定" otherButtonTitles:nil, nil];
04.     [alter show];
05. }

```

效果：



以上是Plain风格的TableView