

# iPhone与iPad应用开发课程 精通iOS开发

## 第六讲 高级UI控件 – 表视图

主讲人：关东升

[eorient@sina.com](mailto:eorient@sina.com)

# 主要知识点

- ◆ UI高级控件表视图
- ◆ 表视图控制器

# 关于表视图

- ◆ iOS中很多应用都使用了表视图，表视图可以分为：
  - 无格式表视图
  - 分段（**Sections**）表视图，而分段表视图又分为：
    - 普通分段表视图
    - 分组分段表视图
    - 索引分段表视图

This is a screenshot of a mobile application's 'Groups' (群组) section. The top status bar shows the carrier '运营商', signal strength, Wi-Fi, and the time '上午3:16'. The app's header is a blue bar with the text '群组'. Below the header is a list of contact groups. The first three items are '全部联系人' (All contacts), 'Friends', and 'Work', each followed by a right-pointing arrow. The remaining space in the screenshot shows several empty rows with horizontal dividers, indicating a scrollable list.

# 分组分段表视图



# 索引分段表视图

运营商	上午3:16	
群组	全部联系人	+
Q 搜索		Q
A		A
John <b>Appleseed</b>		B
		C
B		D
Kate <b>Bell</b>		E
		F
H		G
Anna <b>Haro</b>		H
		I
Daniel <b>Higgins Jr.</b>		J
		K
T		L
David <b>Taylor</b>		M
		N
Z		O
Hank M. <b>Zakroff</b>		P
		Q
		R
		S
		T
		U
		V
		W
		X
		Y
		Z
		#

# 简单表视图(无格式)



# SimpleTableViewController.h

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

@interface SimpleTableViewController : UIViewController
    <UITableViewDataSource, UITableViewDelegate> {
        NSArray *listData;
        NSArray *listImage;
    }
@property (nonatomic, retain) NSArray *listData;
@property (nonatomic, retain) NSArray *listImage;
@end
```

代码参考SimpleTable



# 表视图中的数据源协议

- tableView: numberOfRowsInSection:
- 提供表视图某个分段的行数。
- tableView: cellForRowAtIndexPath:
- 提供表视图单元格所需要的数据。

# 表视图中的委托协议

- tableView: didSelectRowAtIndexPath:
- 该委托方法是表视图单元格选择完成之后触发的方法。

# m文件中的初始化加载方法

```
- (void)viewDidLoad {
    NSArray *array = [[NSArray alloc] initWithObjects:@"A1-南非",@"A2-墨西哥",
        @"B1-阿根廷",@"B2-尼日利亚",@"C1-英格兰",@"C2-美国",
        @"D1-德国",@"D2-澳大利亚",@"E1-荷兰",@"E2-丹麦",
        @"G1-巴西",@"G2-朝鲜",@"H1-西班牙",@"H2-瑞士",nil];
    NSArray *images = [[NSArray alloc] initWithObjects:@"SouthAfrica.png",@"Mexico.png",
        @"Argentina.png",@"Nigeria.png",@"England.png",@"USA.png",
        @"Germany.png",@"Australia.png",@"Holland.png",@"Denmark.png",
        @"Brazil.png",@"NorthKorea.png",@"Spain.png",@"Switzerland.png",nil];
    self.listData = array;
    self.listImage = images;
    [array release];
    [images release];
    [super viewDidLoad];
}
```

# m释放资源

```
- (void)viewDidUnload {  
    [super viewDidUnload];  
    self.listData = nil;  
    self.listImage = nil;  
}  
  
- (void)dealloc {  
    [listData release];  
    [listImage release];  
    [super dealloc];  
}
```

# 实现UITableView数据源方法

```
- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section {
    return [listData count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    static NSString *SimpleCellIdentifier = @"SimpleCellIdentifier";
    UITableViewCell *cell = [tableView
        dequeueReusableCellWithIdentifier:SimpleCellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc]
            initWithStyle:UITableViewCellStyleDefault
            reuseIdentifier:SimpleCellIdentifier] autorelease];
    }
    NSUInteger row = [indexPath row];
    cell.textLabel.text = [listData objectAtIndex:row];
    UIImage *img = [UIImage imageNamed:[listImage objectAtIndex:row]];
    cell.imageView.image = img;
    return cell;
}
```

# 解释

- - (NSInteger)tableView:(UITableView \*)tableView
- numberOfRowsInSection:(NSInteger)section
- 该方法是为表视图提供分段的个数，只有一个分段所以。
- - (UITableViewCell \*)tableView:(UITableView \*)tableView
- cellForRowAtIndexPath:(NSIndexPath \*)indexPath
- 提供表视图单元个所需要的数据。
- static NSString \*SimpleCellIdentifier =  
@"SimpleCellIdentifier";
- 该语句为表视图单元格提供了一个标识，当上面的单元格滚出屏幕，下面的单元格滚入屏幕时候，可以通过判断这个标识是否有可以重用的单元格，如果有则重用，如果没有则创建一个新的。

# 解释

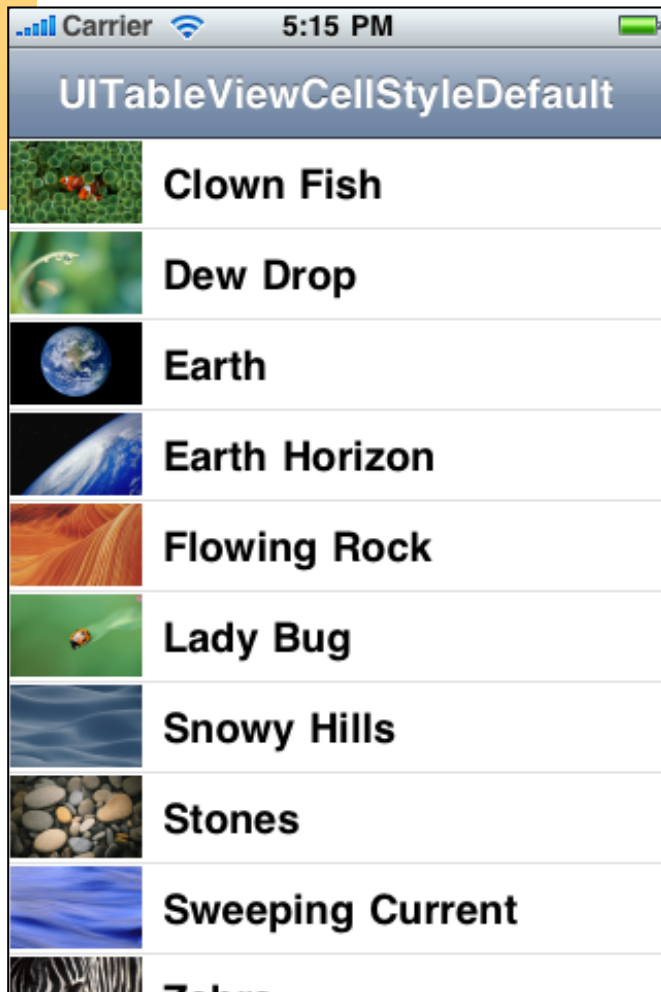
- ◆ 表视图单元格的单元元素：
  - 图片 `cell.imageView.image`
  - 文本标签 `cell.textLabel.text`
- ◆ 详细文本标签 `cell.detailTextLabel.text`



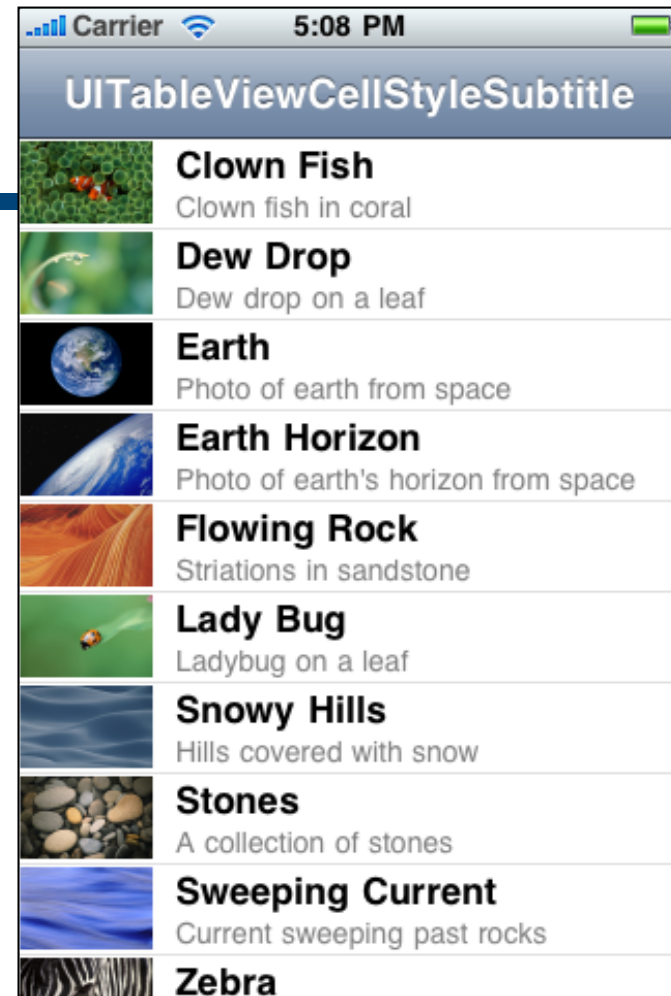
# 单元格样式

- ◆ UITableViewCellStyle枚举类型有成员：
- ◆ UITableViewCellStyleDefault,
- ◆ UITableViewCellStyleValue1,
- ◆ UITableViewCellStyleValue2,
- ◆ UITableViewCellStyleSubtitle

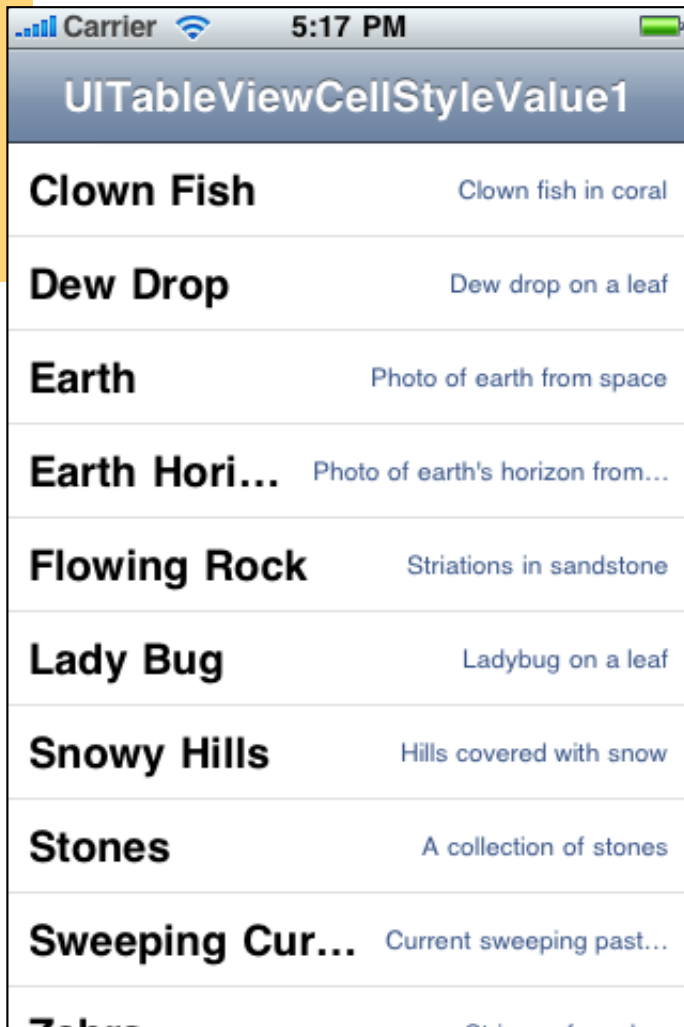




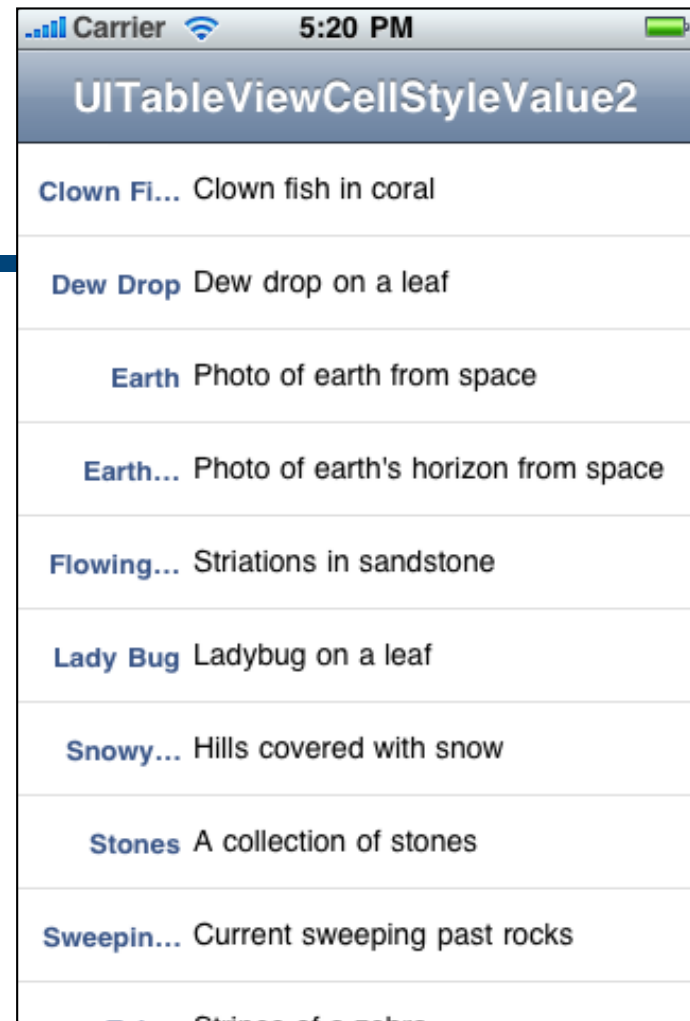
UITableViewCellStyleDefault



UITableViewCellStyleSubtitle



UITableViewCellStyleValue1



UITableViewCellStyleValue2

# 实现UITableView委托方法

```
- (void)tableView:(UITableView *)tableView
    didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    NSUInteger row = [indexPath row];
    NSString *rowValue = [listData objectAtIndex:row];

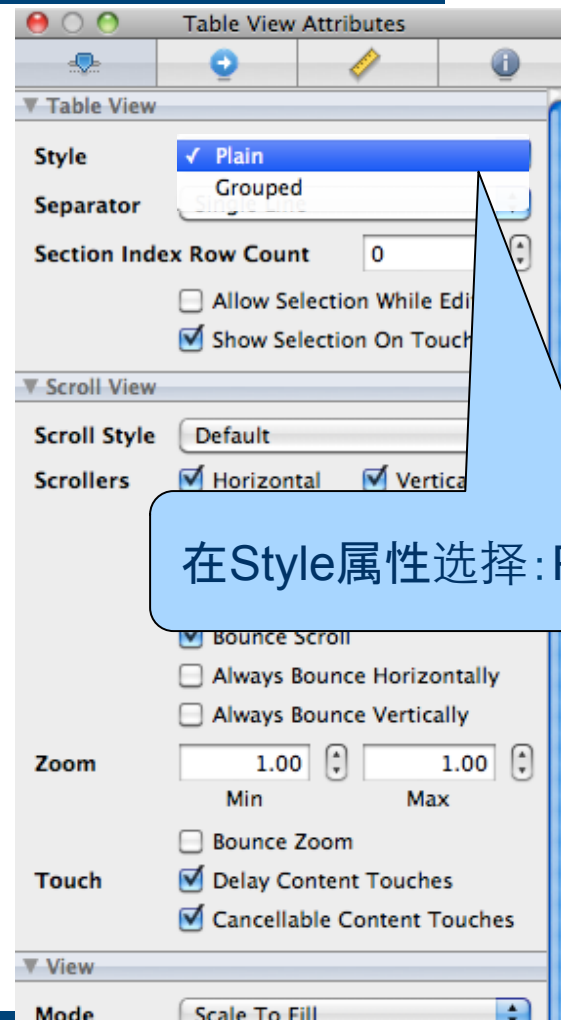
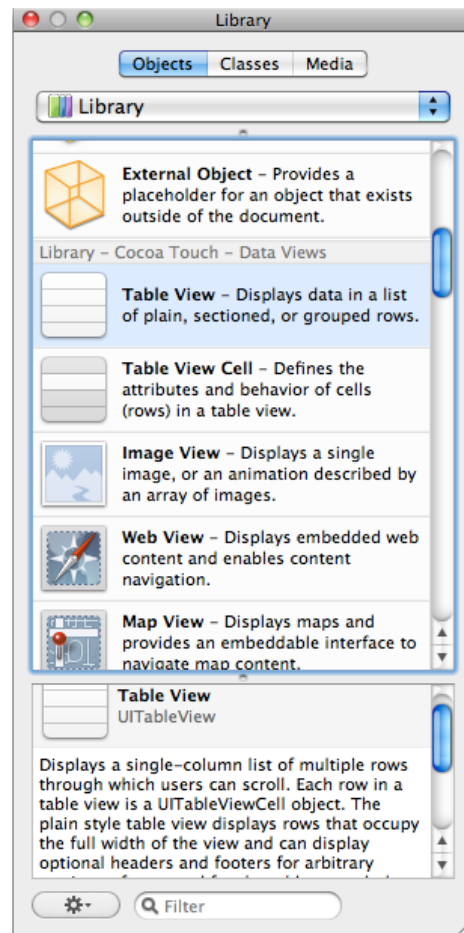
    NSString *message = [[NSString alloc] initWithFormat:@"你选择了%@队。", rowValue];
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"行选择"
                                                         message:message
                                                         delegate:self
                                                         cancelButtonTitle:@"Ok"
                                                         otherButtonTitles:nil];

    [alert show];
    [alert release];
    [message release];
    [tableView deselectRowAtIndexPath:indexPath animated:YES];
}
```

# 解释

- ◆ 该委托方法是选中的单元格时候触发的方法，
- ◆ `[indexPath row]`语句可以获得选中的行号。
- ◆ `[tableView deselectRowAtIndexPath:indexPath animated:YES];`
- ◆ 是可以在表视图选择完成单元格后背景消失，再弹出**alert**对话框。

# 设计NIB文件



# 链接输出口

- ◆ Table View的委托和数据源输出口，需要File's Owner与委托和数据源链接。

# 分段表视图



运营商	下午11:09
A组	
A1-南非	
A2-墨西哥	
A3-乌拉圭	
A4-法国	
B组	
B1-阿根廷	
B2-尼日利亚	
B3-韩国	
B4-希腊	
C组	
C1-英格兰	

# SectionTableViewController.h

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

@interface SectionTableViewController : UIViewController
    <UITableViewDataSource, UITableViewDelegate> {
    NSDictionary *teams;
    NSArray *teamsname;
}
@property (nonatomic,retain) NSDictionary *teams;
@property (nonatomic,retain) NSArray *teamsname;

@end
```

代码参考SectionTable



# 解释

- ◆ UITableViewDataSource, 表视图中的数据源:  
tableView: numberOfRowsInSection:
  - ◆ 提供表视图分段的个数。
- ◆ tableView: cellForRowAtIndexPath:
  - ◆ 提供表视图单元个所需要的数据。
- ◆ UITableViewDelegate, 表视图中的委托:
  - ◆ tableView: didSelectRowAtIndexPath:
    - ◆ 该委托方法是表视图单元个选择之后发生触发

# m文件初始化加载方法

```
- (void)viewDidLoad {  
  
    NSBundle *bundle = [NSBundle mainBundle];  
    NSString *filePath = [bundle pathForResource:@"足球队dictionary"  
                        ofType:@"plist"];  
    NSDictionary *dict = [[NSDictionary alloc] initWithContentsOfFile:filePath];  
    self.teams = dict;  
    [dict release];  
    self.teamsname = [[teams allKeys] sortedArrayUsingSelector:@selector  
(compare:)];  
  
}
```

# 解释

- ◆ 从属性文件中加载数据到NSDictionary对象中，它的key作为表视图的分段的名称，而value部分集合作为表视图单元格。

# 编辑数据属性文件

Key	Type	Value
▼ Root	Dictionary ▴	(8 items)
▼ A组	Array	(4 items)
Item 0	String	A1-南非
Item 1	String	A2-墨西哥
Item 2	String	A3-乌拉圭
Item 3	String	A4-法国
▼ B组	Array	(4 items)
Item 0	String	B1-阿根廷
Item 1	String	B2-尼日利亚
Item 2	String	B3-韩国
Item 3	String	B4-希腊
▶ C组	Array	(4 items)
▶ D组	Array	(4 items)
▶ E组	Array	(4 items)
▶ F组	Array	(4 items)
▶ G组	Array	(4 items)
▶ H组	Array	(4 items)

# m释放资源

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    self.teams = nil;  
    self.teamsname = nil;  
  
}  
  
- (void)dealloc {  
    [teams release];  
    [teamsname release];  
    [super dealloc];  
}
```

# 实现UITableView数据源方法

```
- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section {
    NSString *name = [teamsname objectAtIndex:section];
    NSArray *team = [teams objectForKey:name];
    return [team count];
}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
    return [teamsname count];
}

- (NSString *)tableView:(UITableView *)tableView
titleForHeaderInSection:(NSInteger)section {
    NSString *name = [teamsname objectAtIndex:section];
    return name;
}
```

# 解释

- ◆ - (NSInteger)tableView:(UITableView \*)tableView  
numberOfRowsInSection:(NSInteger)section
- ◆ 该方法是为表视图提供某个分段中单元格个数。
- ◆ - (NSInteger)numberOfSectionsInTableView:  
(UITableView \*)tableView
- ◆ 提供表视图中分段的个数。
- ◆ - (NSString \*)tableView:(UITableView \*)tableView  
titleForHeaderInSection:(NSInteger)section
- ◆ 提供表视图中分段的Header信息。

# 接上实现UITableView数据源方法

```
- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    NSUInteger section = [indexPath section];
    NSUInteger row = [indexPath row];
    NSString *name= [teamsname objectAtIndex:section];
    NSArray *team = [teams objectForKey:name];

    static NSString *CellIdentifier = @"CellIdentifier";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault
                                         reuseIdentifier:CellIdentifier] autorelease];
    }

    cell.textLabel.text = [team objectAtIndex:row];
    return cell;
}
```



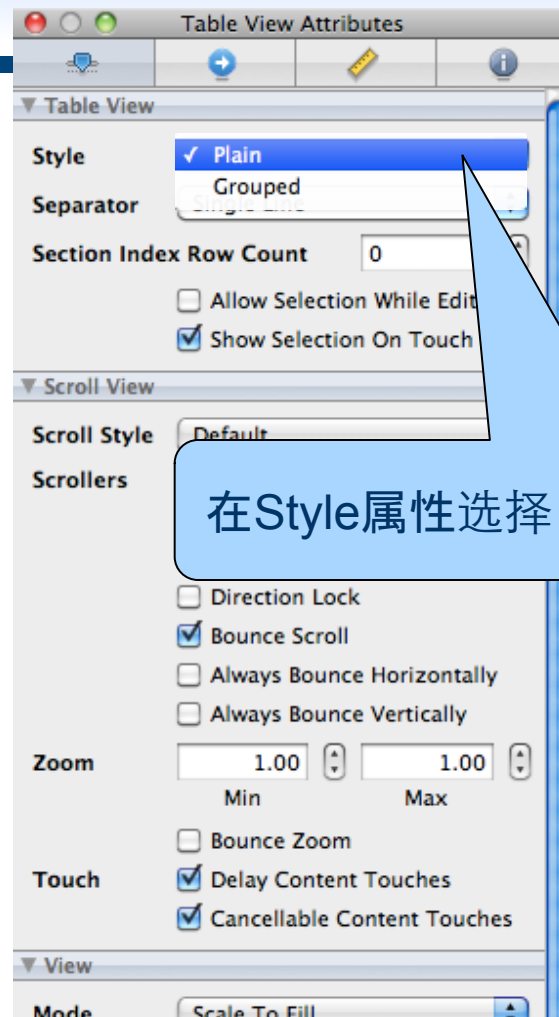
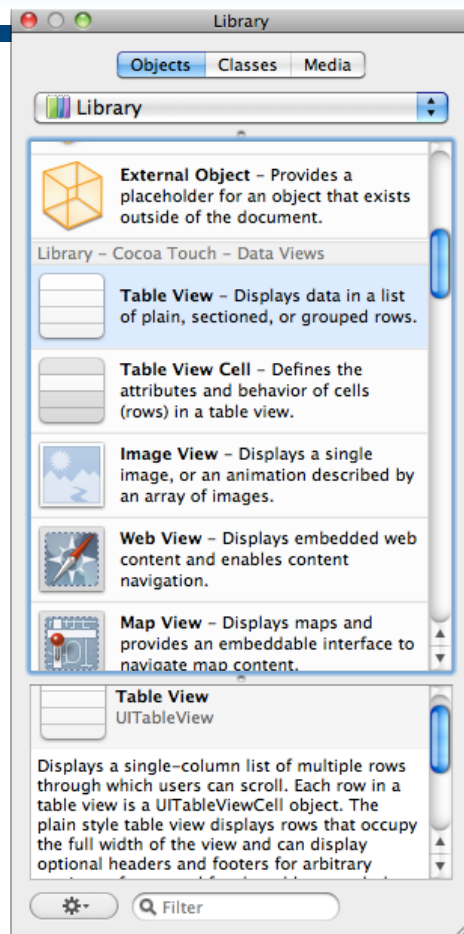
# 实现UITableView委托方法

```
-(void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {  
    NSUInteger section = [indexPath section];  
    NSUInteger row = [indexPath row];  
    NSString *name = [teamsname objectAtIndex:section];  
    NSArray *team = [teams objectForKey:name];  
    NSString *selectedteam = [team objectAtIndex:row];  
  
    NSString *message = [[NSString alloc] initWithFormat:@"你选择了%@队。",  
        selectedteam];  
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"行选择"  
        message:message  
        delegate:self  
        cancelButtonTitle:@"Ok"  
        otherButtonTitles:nil];  
  
    [message release];  
    [alert show];  
    [alert release];  
    [tableView deselectRowAtIndexPath:indexPath animated:YES];  
}
```

# 解释

- ◆ 该委托方法是选中的单元格时候触发的方法，
- ◆ `[indexPath row]`语句可以获得选中的行号。
- ◆ `[tableView deselectRowAtIndexPath:indexPath animated:YES];`
- ◆ 是可以在表试图选择完成单元格后背景消失，再弹出**alert**对话框。

# 设计NIB文件



# 链接输出口

- ◆ Table View的委托和数据源输出口，需要File's Owner与委托和数据源链接。

# 分组分段表视图

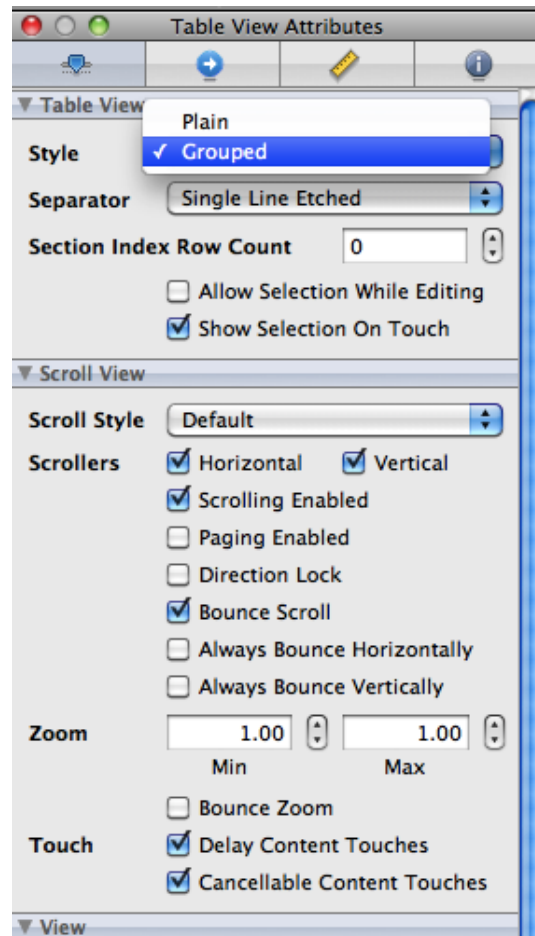
- ◆ 由普通分段表变成分组分段表很简单，修改属性就可以了



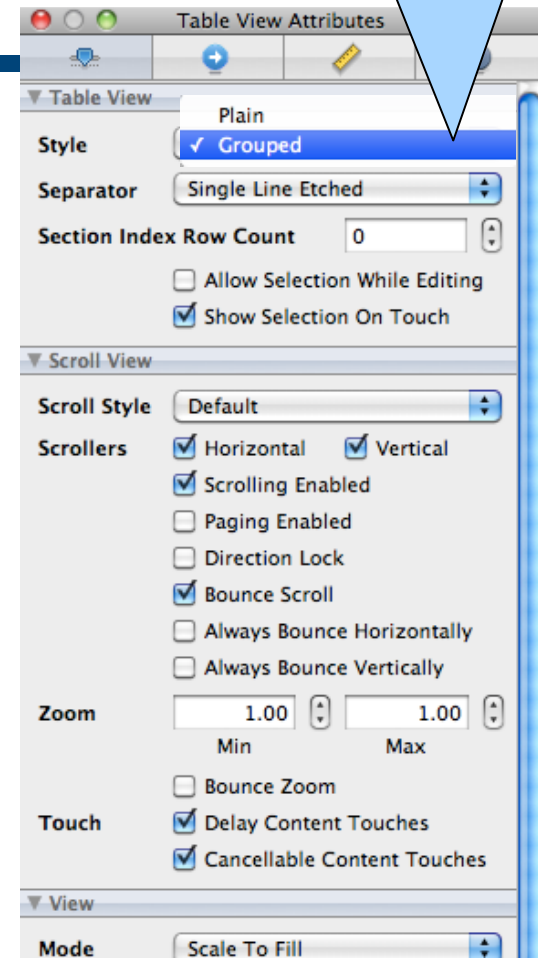
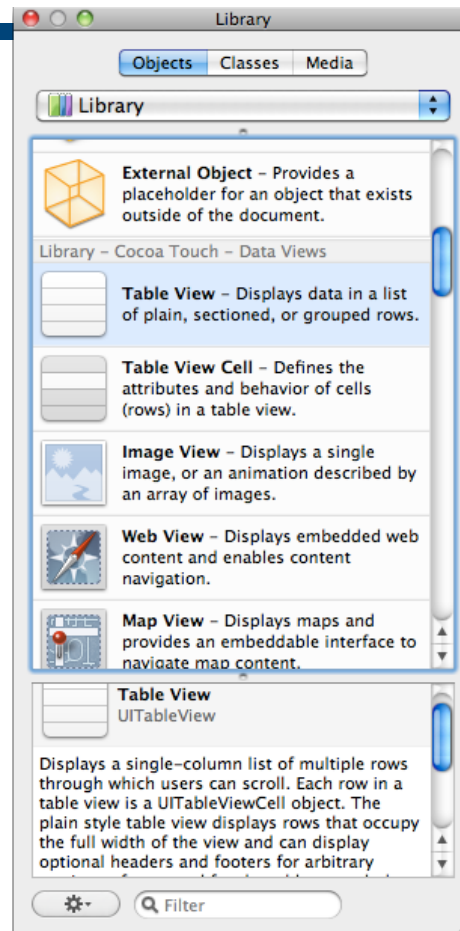
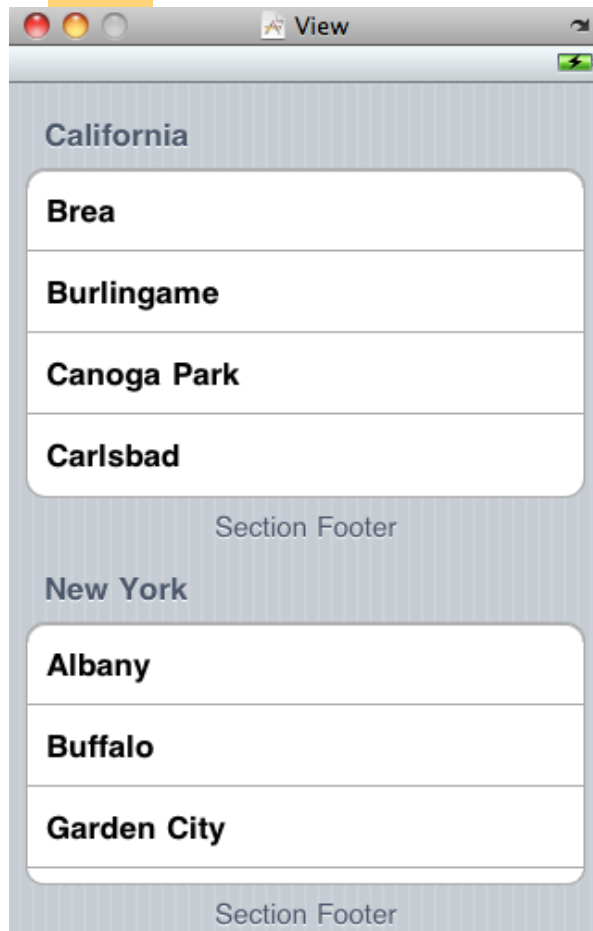
运营商	下午11:09
A组	
A1-南非	
A2-墨西哥	
A3-乌拉圭	
A4-法国	
B组	
B1-阿根廷	
B2-尼日利亚	
B3-韩国	
B4-希腊	

# 修改分组分段

- ◆ 在IB中打开表视图属性设置框，在Style项目选择Grouped。



在Style属性选择: Grouped



# 索引表视图

- 索引表就是在右边增加一个索引列，如果按照归类它应该属于分段表范围。

代码参考

**SectionIndexingTable**

运营商 下午10:52	
C组	A组
C1-英格兰	
C2-美国	B组
C3-阿尔及利亚	C组
C4-斯洛文尼亚	
D组	D组
D1-德国	E组
D2-澳大利亚	
D3-塞尔维亚	F组
D4-加纳	G组
E组	
E1-荷兰	H组



# 索引表实现

- ◆ 索引表实现很简单只是需要在分段表的代码基础上增加一个数据源（UITableViewController）实现方法就可以了：

```
-(NSArray *) sectionIndexTitlesForTableView: (UITableView *)  
tableView {  
    return teamsname;  
}
```
- ◆ 这个方法是为表视图提供索引所需要的数据集集合。

# 实现搜索栏



代码参考  
SectionSearchTable

# SectionTableViewController.h

```
@interface SectionTableViewController : UIViewController<UITableViewDelegate,
UITableViewDataSource,UISearchBarDelegate> {
    NSMutableDictionary *allteams;
    NSMutableDictionary *teams;
    NSArray *teamsname;
}

@property (nonatomic,retain) NSMutableDictionary *teams;
@property (nonatomic,retain) NSMutableDictionary *allteams;
@property (nonatomic,retain) NSArray *teamsname;

-(void)resetSearch;
```

# m文件中的初始化方法

```
- (void)viewDidLoad {
```

```
    NSBundle *bundle = [NSBundle mainBundle];
```

```
    NSString *filePath = [bundle pathForResource:@"足球队dictionary"  
ofType:@"plist"];
```

```
    NSMutableDictionary *dict = [[NSDictionary alloc] initWithContentsOfFile:filePath];
```

```
    self.allteams = dict;
```

```
    [dict release];
```

```
    [self resetSearch];
```

```
}
```

# m文件中的resetSearch方法

```
-(void)resetSearch {  
  
    self.teams = self.allteams ;  
    NSMutableArray *keysArray = [[NSMutableArray alloc] init];  
    [keysArray addObjectsFromArray:[self.teams allKeys]  
sortedArrayUsingSelector:@selector(compare:)];  
    self.teamsname = keysArray;  
    [keysArray release];  
  
}
```

# m释放资源

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    self.teams = nil;  
    self.teamsname = nil;  
}  
  
- (void)dealloc {  
    [teams release];  
    [teamsname release];  
    [super dealloc];  
}
```

# 实现UITableView数据源方法

```
- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section {
    NSString *name = [teamsname objectAtIndex:section];
    NSArray *team = [teams objectForKey:name];
    return [team count];
}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
    return [teamsname count];
}

- (NSString *)tableView:(UITableView *)tableView
titleForHeaderInSection:(NSInteger)section {
    NSString *name = [teamsname objectAtIndex:section];
    return name;
}
```

# 接上实现UITableView数据源方法

```
- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    NSUInteger section = [indexPath section];
    NSUInteger row = [indexPath row];
    NSString *name= [teamsname objectAtIndex:section];
    NSArray *team = [teams objectForKey:name];

    static NSString *CellIdentifier = @"CellIdentifier";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault
                                         reuseIdentifier:CellIdentifier] autorelease];
    }

    cell.textLabel.text = [team objectAtIndex:row];
    return cell;
}
```



# 接上实现UITableView数据源方法

```
/*
```

```
    增加索引
```

```
*/
```

```
-(NSArray *) sectionIndexTitlesForTableView: (UITableView *)
```

```
tableView {
```

```
    return teamsname;
```

```
}
```

# 实现UITableView委托方法

```
-(void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {
    NSUInteger section = [indexPath section];
    NSUInteger row = [indexPath row];
    NSString *name = [teamsname objectAtIndex:section];
    NSArray *team = [teams objectForKey:name];
    NSString *selectedteam = [team objectAtIndex:row];

    NSString *message = [[NSString alloc] initWithFormat:@"你选择了%@队。",
                                                             selectedteam];
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"行选择"
                                                         message:message
                                                         delegate:self
                                                         cancelButtonTitle:@"Ok"
                                                         otherButtonTitles:nil];

    [message release];
    [alert show];
    [alert release];
    [tableView deselectRowAtIndexPath:indexPath animated:YES];
}
```

# Search Bar 委托方法

```
- (void)searchBar:(UISearchBar *)searchBar textDidChange:(NSString *)searchText {
```

```
    if ([searchText length] == 0) {  
        [self resetSearch];  
        return;  
    }
```

```
    NSMutableDictionary *dict = [[NSMutableDictionary alloc] init];
```

```
    for (NSString *key in self.allTeams) {
```

```
        NSMutableArray *array = [allTeams valueForKey:key];
```

```
        NSMutableArray *newTeams = [[NSMutableArray alloc] init];
```

```
        for (NSString *teamName in array) {
```

```
            if ([teamName rangeOfString: searchText  
                                options:NSCaseInsensitiveSearch].location != NSNotFound) {  
                [newTeams addObject:teamName];  
            }
```

```
        }
```

```
        if ([newTeams count] > 0) {
```

```
            [dict setObject:newTeams forKey:key];  
        }
```

```
        [newTeams release];
```

```
    }
```

# Search Bar 委托方法

```
self.teamsname = [[dict allKeys] sortedArrayUsingSelector:@selector(compare:)];  
self.teams = dict;  
[dict release];  
}  
  
- (void)searchBarCancelButtonClicked:(UISearchBar *)searchBar {  
    [self resetSearch];  
}
```

# 解释

- (void)searchBar:(UISearchBar \*)searchBar  
textDidChange:(NSString \*)searchText
  - ◆ 搜索栏中的文本发生变化的时候，触发这个方法。
- (void)searchBarCancelButtonClicked:(UISearchBar \*)  
searchBar
  - ◆ 点击搜索栏的取消按钮时候调用。

# 视图设计

