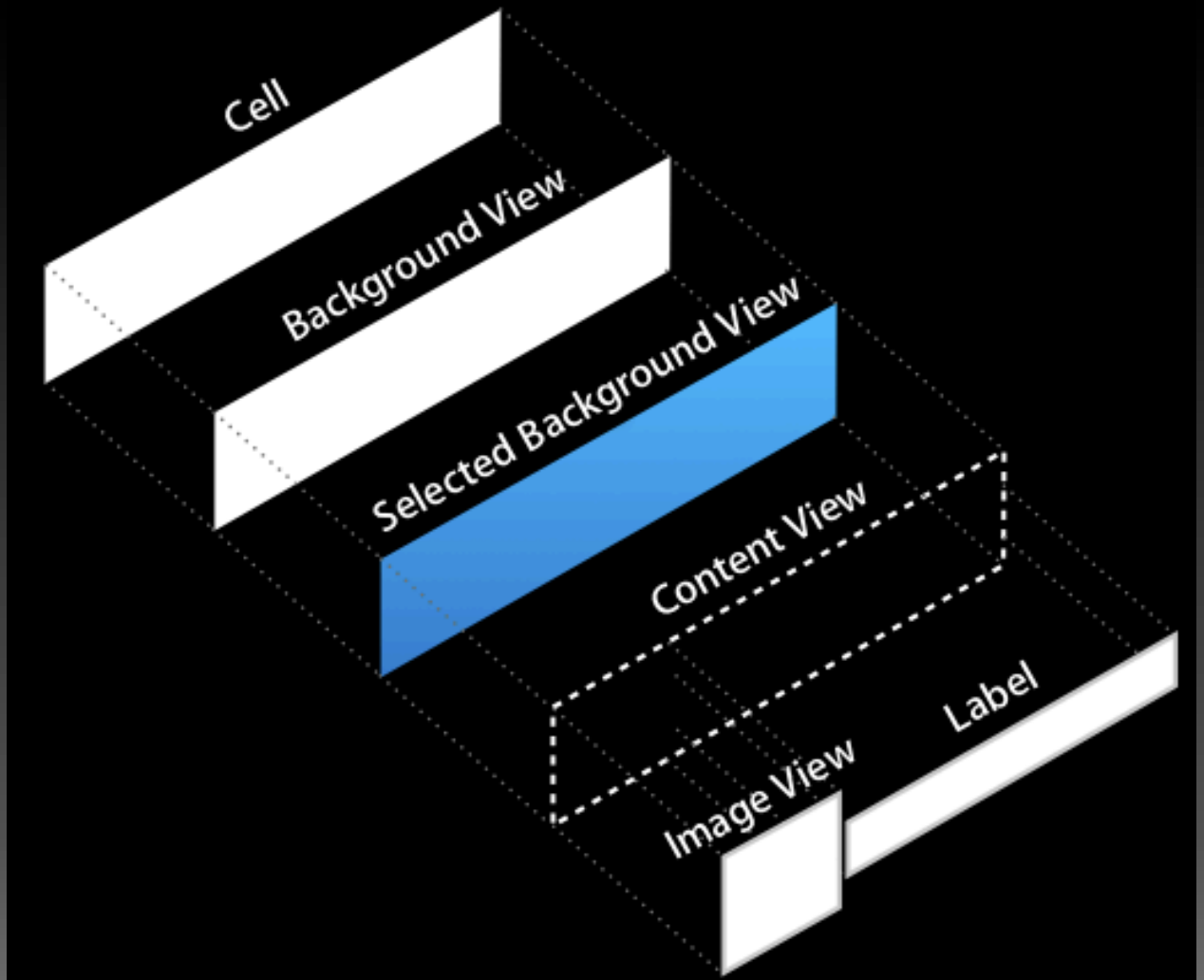
Table View Search

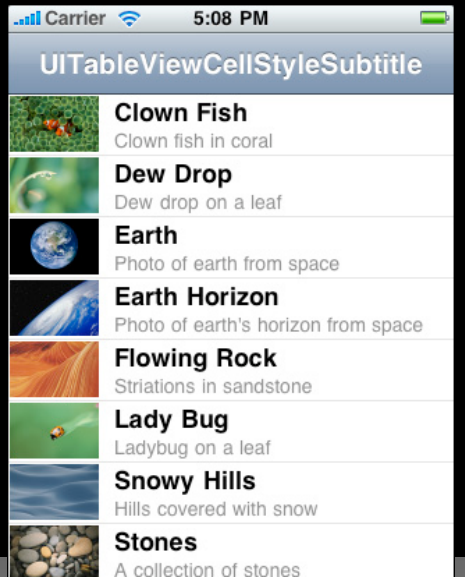
Trinh Minh Cuong
cuong@techmaster.vn

CUTOMIZE TABLE VIEW CELL



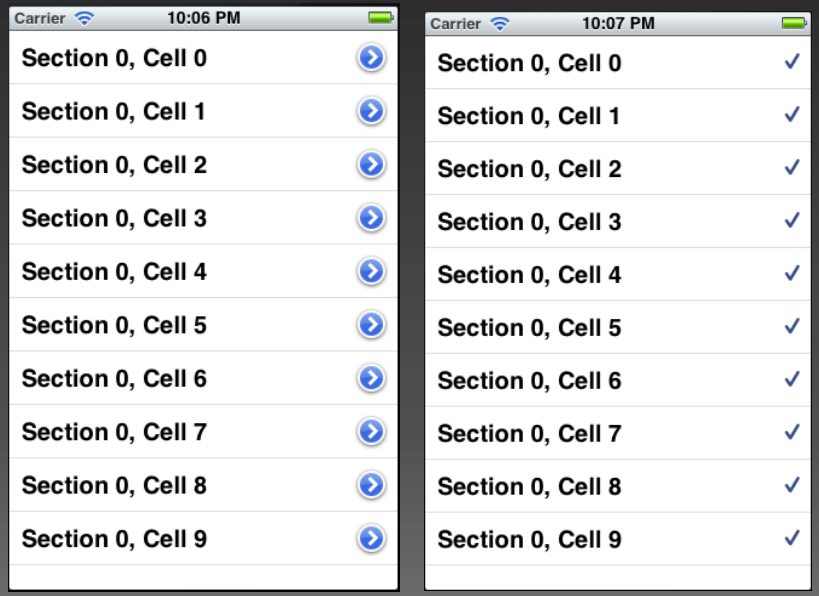
http://techmaster.vn

```
- (UITableViewCell *)tableView:(UITableView *)tableView
- cellForRowAtIndexPath:(NSIndexPath *)indexPath {

    UITableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:@"MyIdentifier"];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleSubtitle
reuseIdentifier:@"MyIdentifier"] autorelease];
        cell.selectionStyle = UITableViewCellSelectionStyleNone;
    }
NSDictionary *item = (NSDictionary *)[self.content objectAtIndex:indexPath.row];

cell.textLabel.text = [item objectForKey:@"Title"];
cell.detailTextLabel.text = [item objectForKey:@"subTitle"];

NSString *path = [[NSBundle mainBundle] pathForResource:[item
objectForKey:@"imageKey"] ofType:@"png"];
UIImage *theImage = [UIImage imageWithContentsOfFile:path];

cell.imageView.image = theImage;

return cell;
}
```



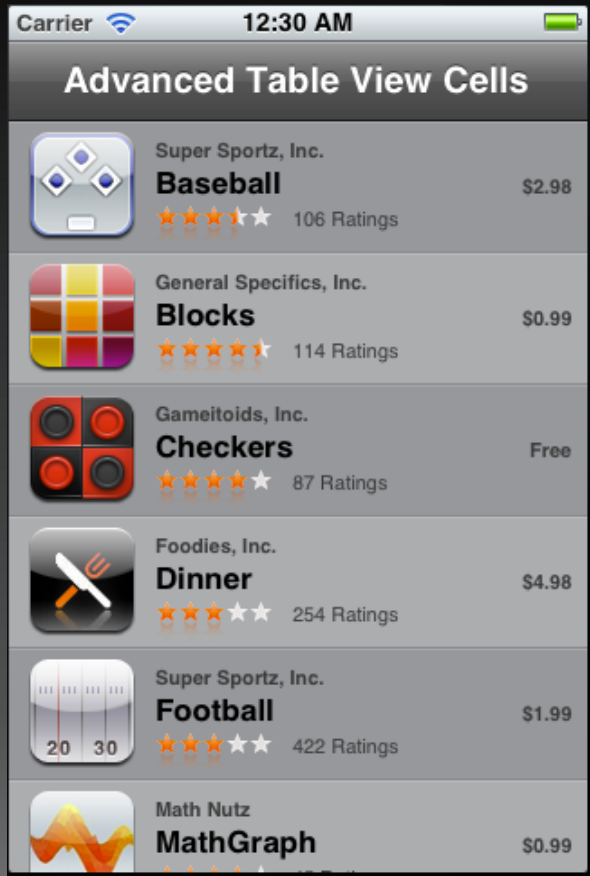
```
result.accessoryType = UITableViewCellAccessoryDetailDisclosureButton;
```

```
typedef enum {
    UITableViewCellAccessoryNone,
    UITableViewCellAccessoryDisclosureIndicator,
    UITableViewCellAccessoryDetailDisclosureButton,
    UITableViewCellAccessoryCheckmark
} UITableViewCellAccessoryType;
```

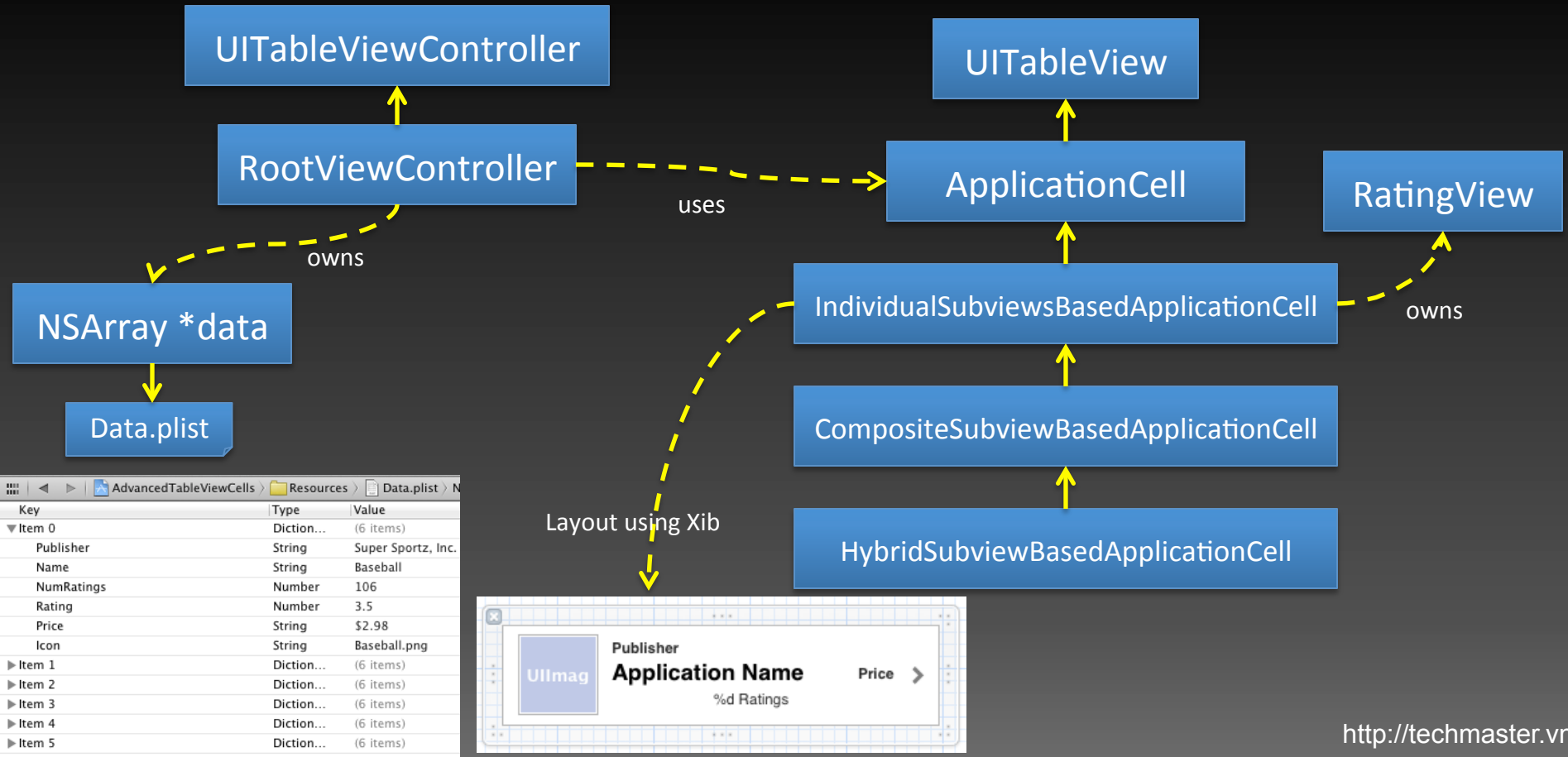


```
result.textLabel.text = [NSString stringWithFormat:@"Section %ld, Cell
%ld",
                                (long)indexPath.section,
                                (long)indexPath.row];

UIButton *button = [UIButton buttonWithType:UIButtonTypeRoundedRect];
button.frame = CGRectMake(0.0f, 0.0f, 150.0f, 25.0f);

[button setTitle:@"Expand"
        forState:UIControlStateNormal];

[button addTarget:self
           action:@selector(performExpand:)
 forControlEvents:UIControlEventTouchUpInside];

result.accessoryView = button;

return result;
```

ADVANCED CUSTOMED TABLE VIEW CELL



Advanced Customed Table View Cell



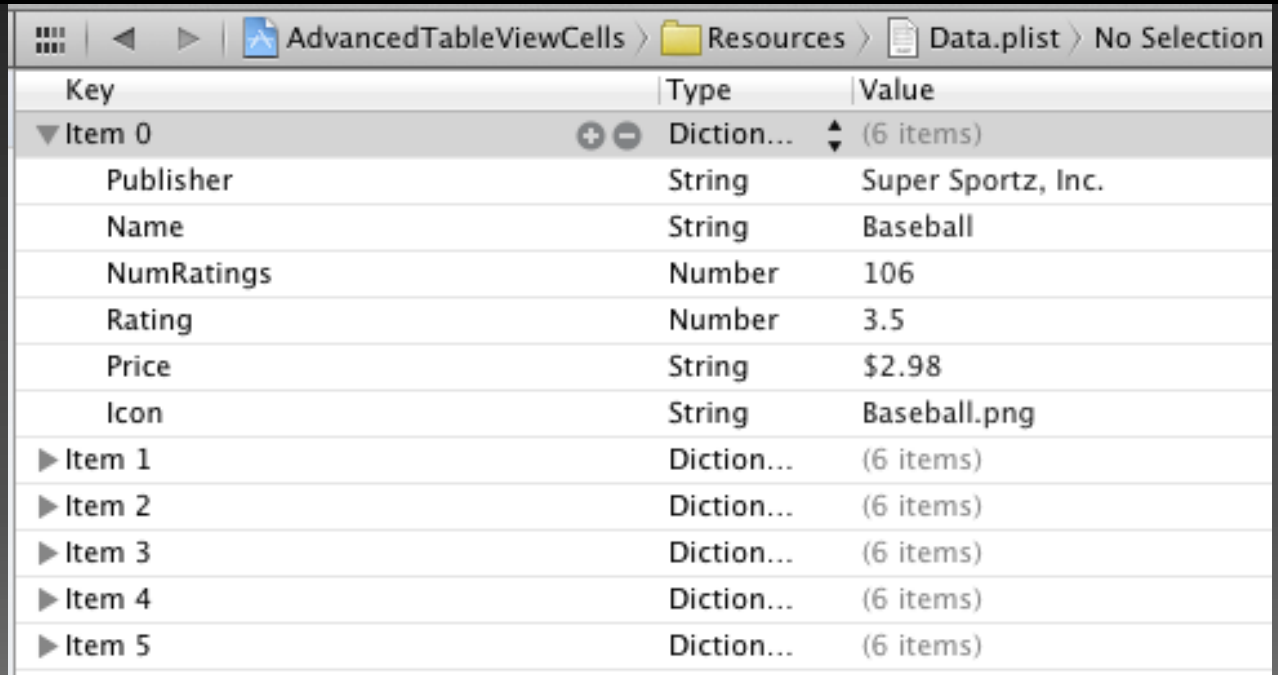
```
[self.cellNib instantiateWithOwner:self options:nil];
cell = tmpCell;
self.tmpCell = nil;
```

1- Load Nib
2- Associate loaded Nib with tmpCell
3- Assign tmpCell to each cell

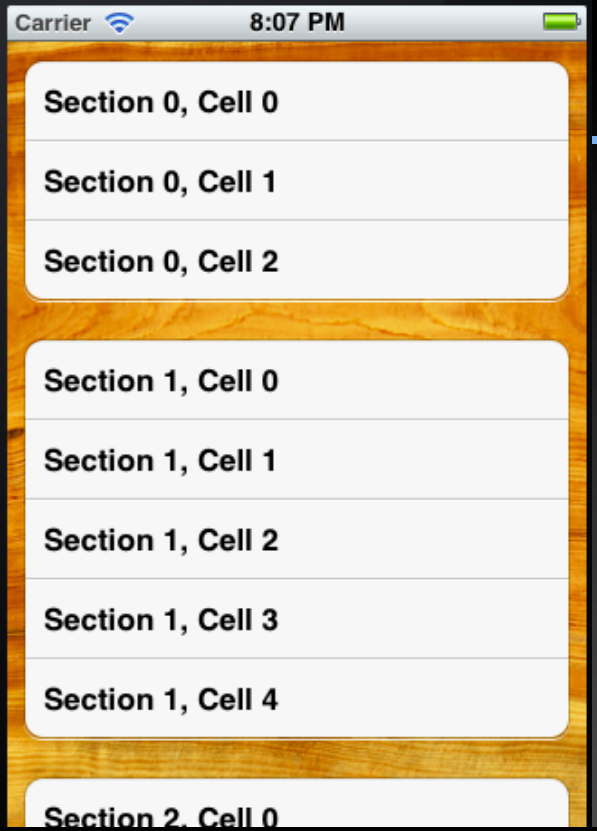
HOW TO LOAD DATA FROM PLIST

Load data from plist to model

```
NSString *dataPath = [[NSBundle mainBundle]
pathForResource:@"Data" ofType:@"plist"];

self.data = [NSArray arrayWithContentsOfFile:dataPath];
```



Customize backgroud

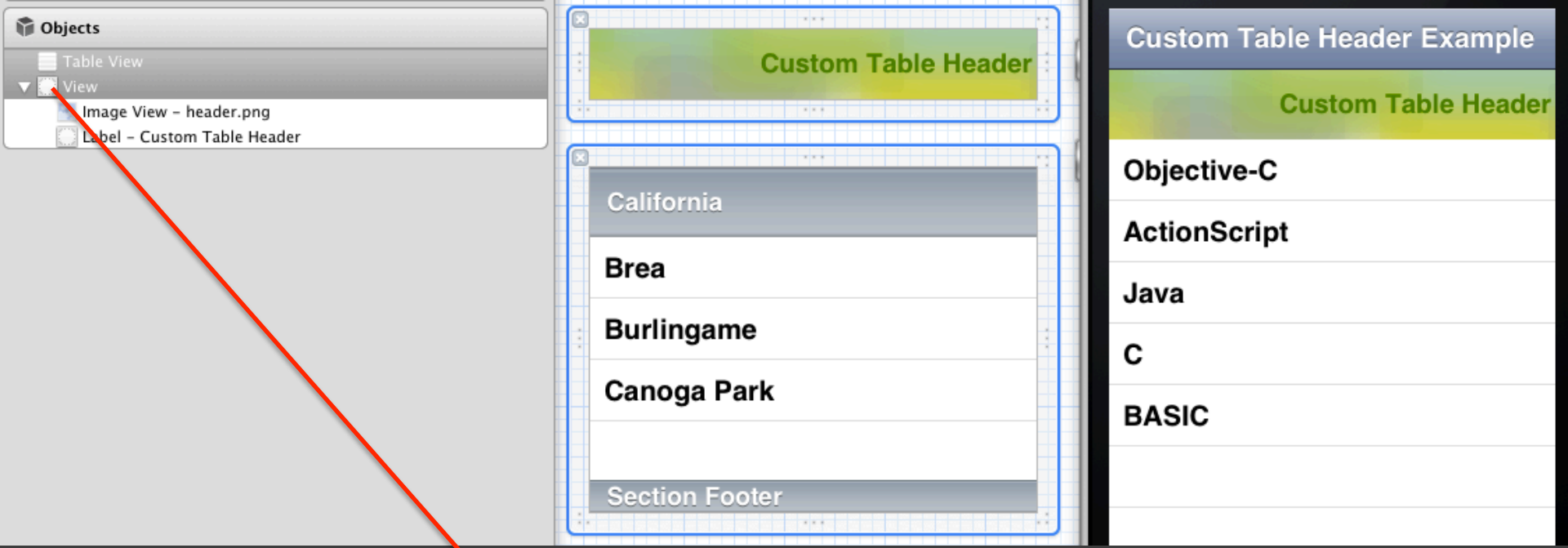



```
UIImage *backgroundImage = [UIImage mageNamed:@"woodgrain.jpg"];

self.myTableView.backgroundView = [[UIImageView alloc]
    initWithImage:backgroundImage];
```

Custom header section

Header and Footer of Sections

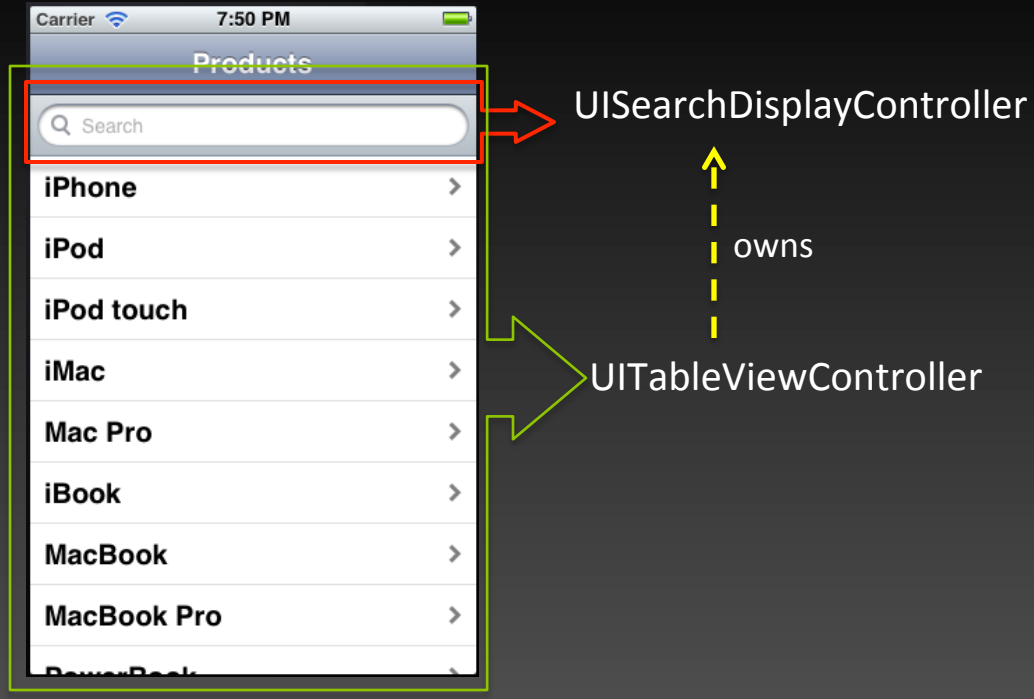
- viewForHeaderInSection:
- viewForFooterInSection:
- heightForHeaderInSection:
- heightForFooterInSection:




```
@interface RootViewController : UITableViewController {
        IBOutlet UIView *headerView;
        NSArray *tableData;
}

@property (nonatomic, retain) UIView *headerView;
@property (nonatomic, retain) NSArray *tableData;
```

SEARCH IN TABLE VIEW

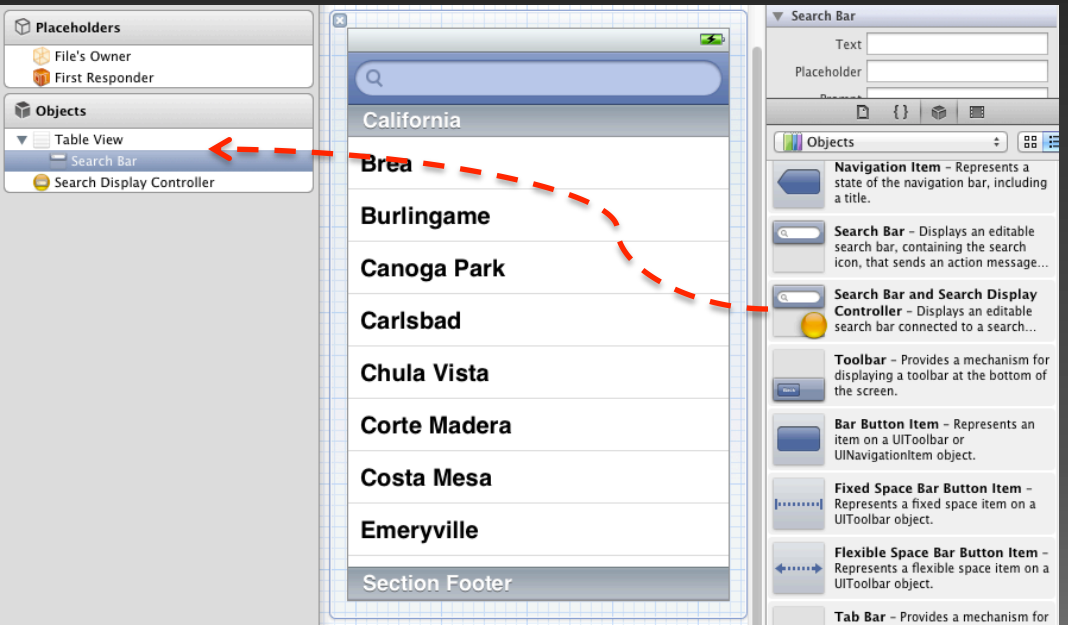
Search for UITableViewController

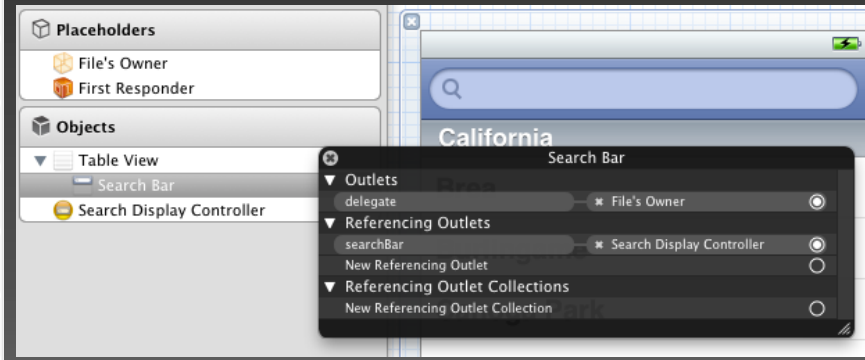


In fact, UITableViewController inherits UIViewController.
UIViewController has pointer to UISearchDisplayController

Steps

- In IB, drag SearchBar and Display Controller to TableView
- Make ViewController to conform UISearchDisplayDelegate, UISearchBarDelegate





UISearchBar Class Reference

Inherits from	UIView : UIResponder : NSObject
Conforms to	NSCoding (UIView) UIAppearance (UIView) UIAppearanceContainer (UIView) NSObject (NSObject)
Framework	/System/Library/Frameworks/UIKit.framework
Availability	Available in iOS 2.0 and later.
Declared in	UISearchBar.h

ster.vn

UISearchDisplayController Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/UIKit.framework
Availability	Available in iOS 3.0 and later.
Declared in	UISearchDisplayController.h
Related sample code	TableSearch

A search display controller manages display of a search bar and a table view that displays the results of a search of data managed by another view controller.

You initialize a search display controller with a search bar and a view controller responsible for managing the original content to be searched.

When the user starts a search, the search display controller is responsible for superimposing the search interface over the original view controller's view and showing the search results.

UISearchDisplayDelegate

Search State Change
- searchDisplayControllerWillBeginSearch:
- searchDisplayControllerDidBeginSearch:
- searchDisplayControllerWillEndSearch:
- searchDisplayControllerDidEndSearch:

Loading and Unloading the Table View
- searchDisplayController:didLoadSearchResultsTableView:
- searchDisplayController:willUnloadSearchResultsTableView:

Showing and Hiding the Table View
- searchDisplayController:willShowSearchResultsTableView:
- searchDisplayController:didShowSearchResultsTableView:
- searchDisplayController:willHideSearchResultsTableView:
- searchDisplayController:didHideSearchResultsTableView:

Responding to Changes in Search Criteria
- searchDisplayController:shouldReloadTableForSearchString:
- searchDisplayController:shouldReloadTableForSearchScope:

Insert, Delete, Move Rows & Selections

Homework

- Create weather forecast GUI like this
- Create searchable table view of product list: product photo thumbnail, product name, short description, price and rating

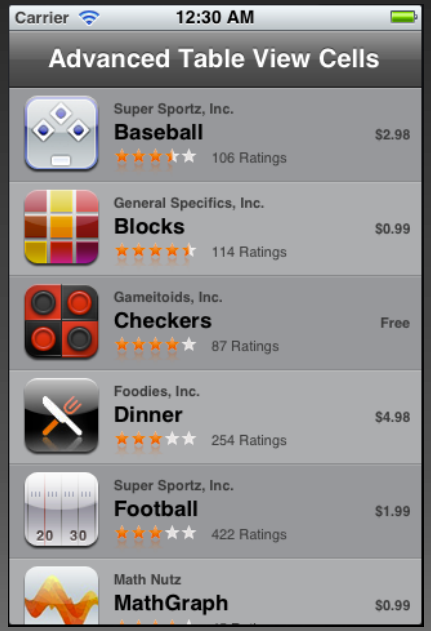


Table Badget

- http://www.tuaw.com/2010/01/07/iphone-devsugar-simple-table-badges/