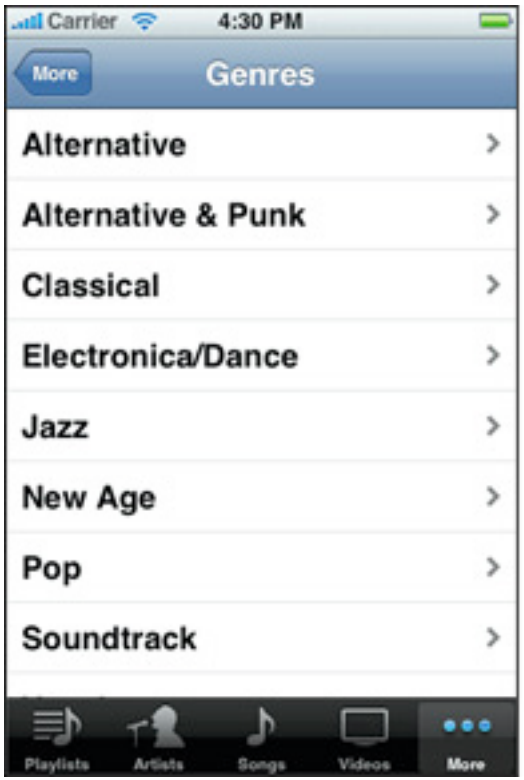
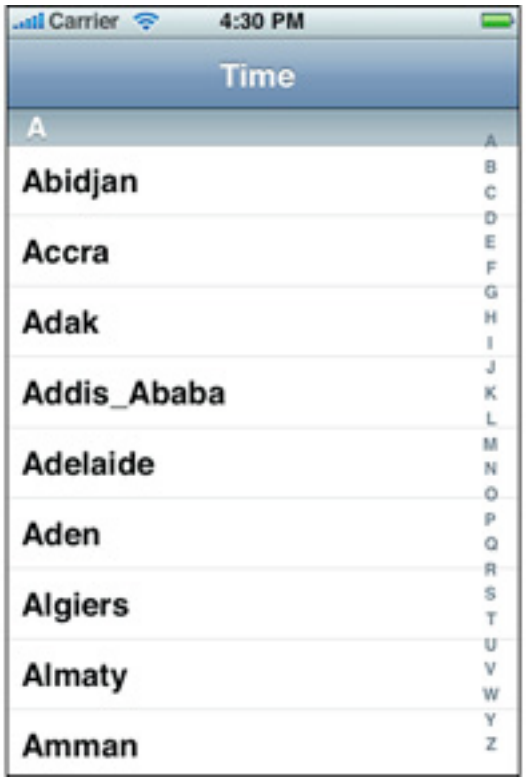
iPhone Game Development

UITableView, aka Brainfreeze without the ice...

plain



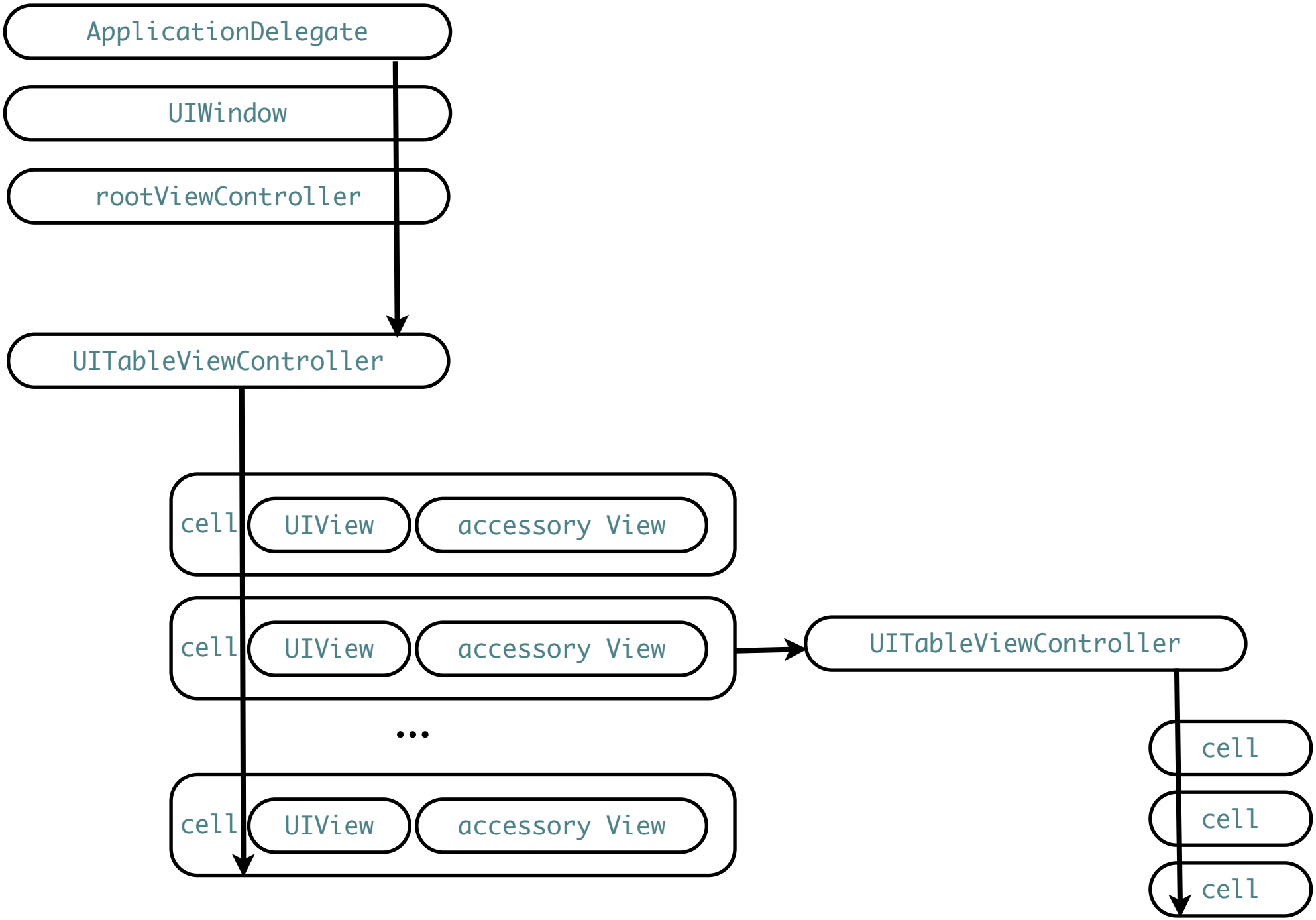
plain, section



grouped

Table View programming guide for iOS, seriously:

http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/TableView_iPhone/AboutTableViewsiPhone/AboutTableViewsiPhone.html#//apple_ref/doc/uid/TP40007451


ApplicationDelegate
UIWindow
rootViewController
UITableViewController
cell
UIView
accessory View
cell
UIView
accessory View
...
cell
UIView
accessory View
UITableViewController
cell
cell
cell

application Delegate:

```
self.window.rootViewController=[[myTableViewController alloc]
initWithStyle:UITableViewStylePlain];
```

myTableViewController is the delegate, and thus confirms to UITableViewDelegate:

Configuring Rows for the Table View
- tableView:heightForRowAtIndexPath:
- tableView:indentationLevelForRowAtIndexPath:
- tableView:willDisplayCell:forRowAtIndexPath:

Managing Selections
- tableView:willSelectRowAtIndexPath:
- tableView:didSelectRowAtIndexPath:
- tableView:willDeselectRowAtIndexPath:
- tableView:didDeselectRowAtIndexPath:

Managing Accessory Views
Modifying the Header and Footer of Sections
Editing Table Rows
Copying and Pasting Row Content

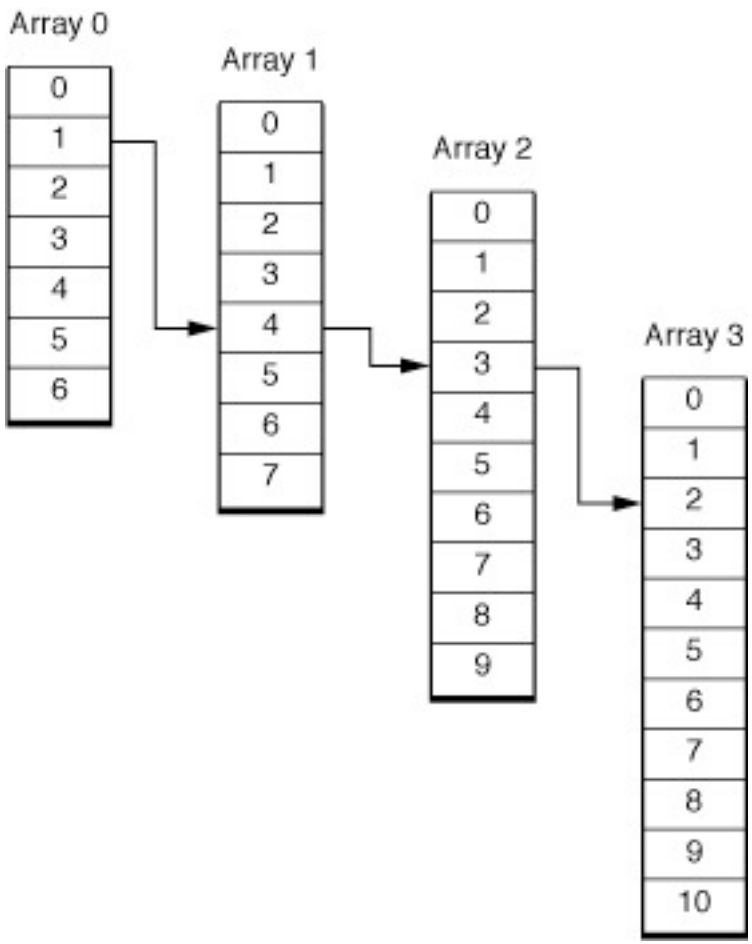
BUT also conforms to UITableViewDataSource:

Configuring a Table View
- tableView:cellForRowAtIndexPath: required method
- numberOfSectionsInTableView:
- tableView:numberOfRowsInSection: required method
- sectionIndexTitlesForTableView:
- tableView:sectionForSectionIndexTitle:atIndex:
- tableView:titleForHeaderInSection:
- tableView:titleForFooterInSection:

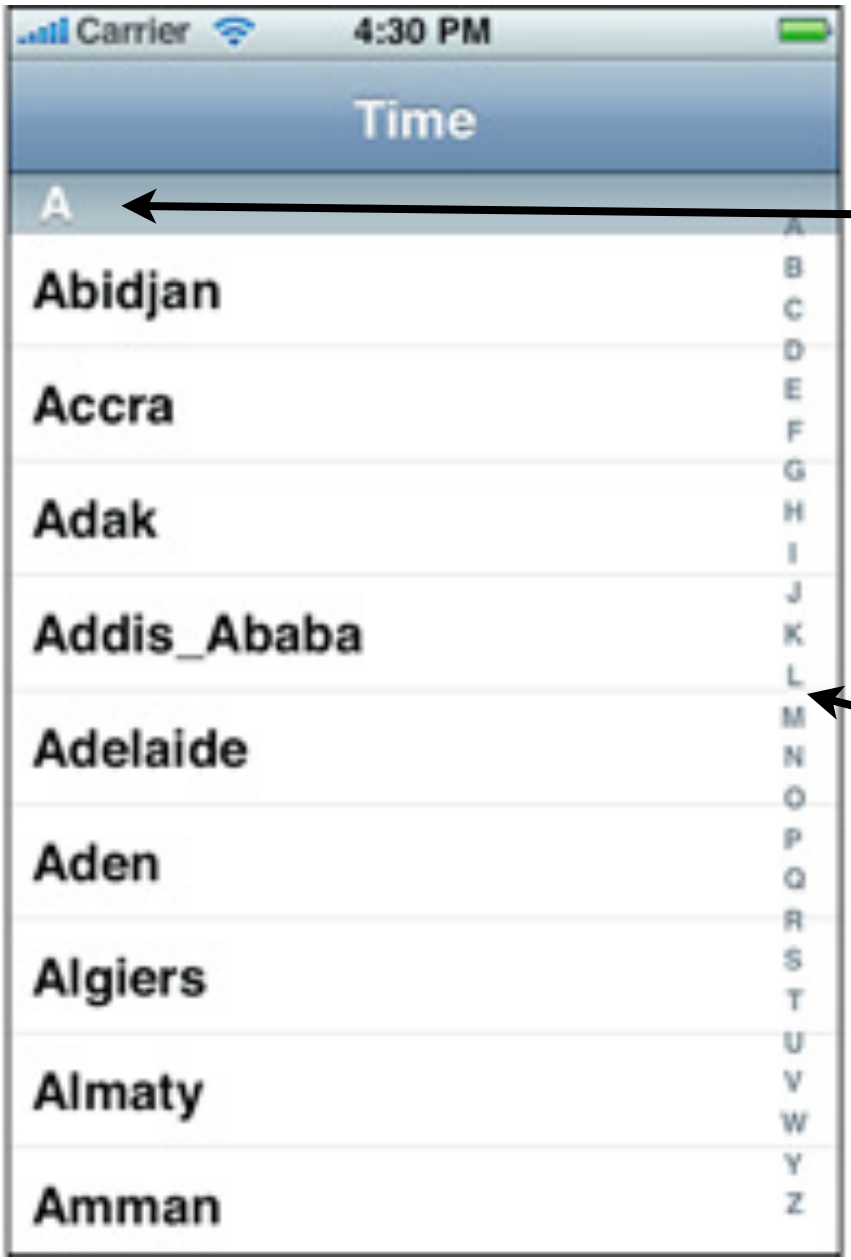
```
- (UITableViewCell *)tableView:(UITableView *)tableView
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView
                    dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc]
                    initWithStyle:UITableViewCellStyleDefault
                    reuseIdentifier:CellIdentifier] autorelease];
    }
    // Configure the cell...



    return cell;
}
```

```
- (UITableViewCell *)tableView:(UITableView *)tableView
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView
                    dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc]
                    initWithStyle:UITableViewCellStyleDefault
                    reuseIdentifier:CellIdentifier] autorelease];
    }
    // Configure the cell...

   [cell.textLabel setText:
          [NSString stringWithFormat:
          @"%i %i",indexPath.section,indexPath.row]];

    return cell;
}
```

NSIndexPath != NSIndexPath for tables



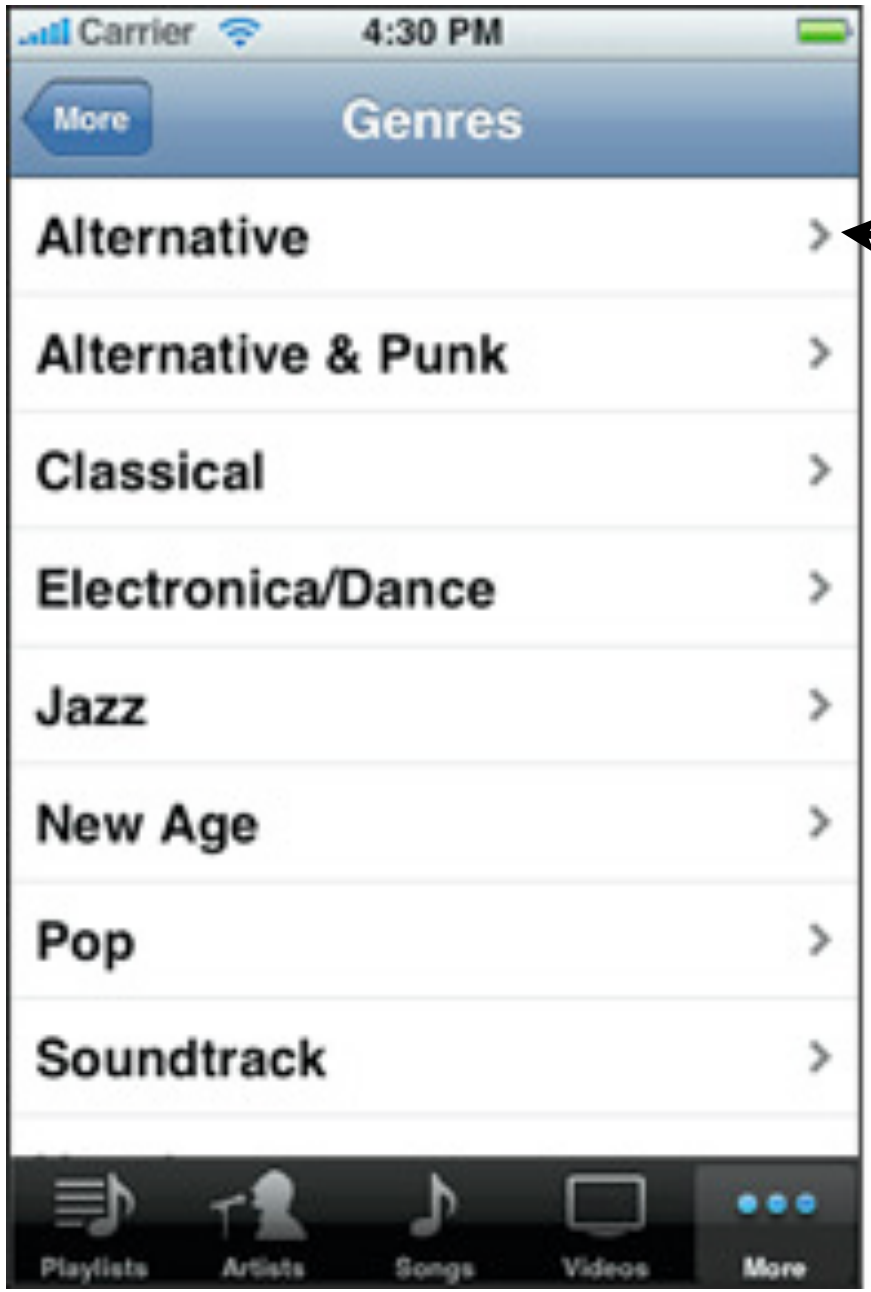
Creating an Index Path Object
+ indexPathForRow:inSection:
Getting the Row and Section Indexes
  row property
  section property



title for header in section

section index titles

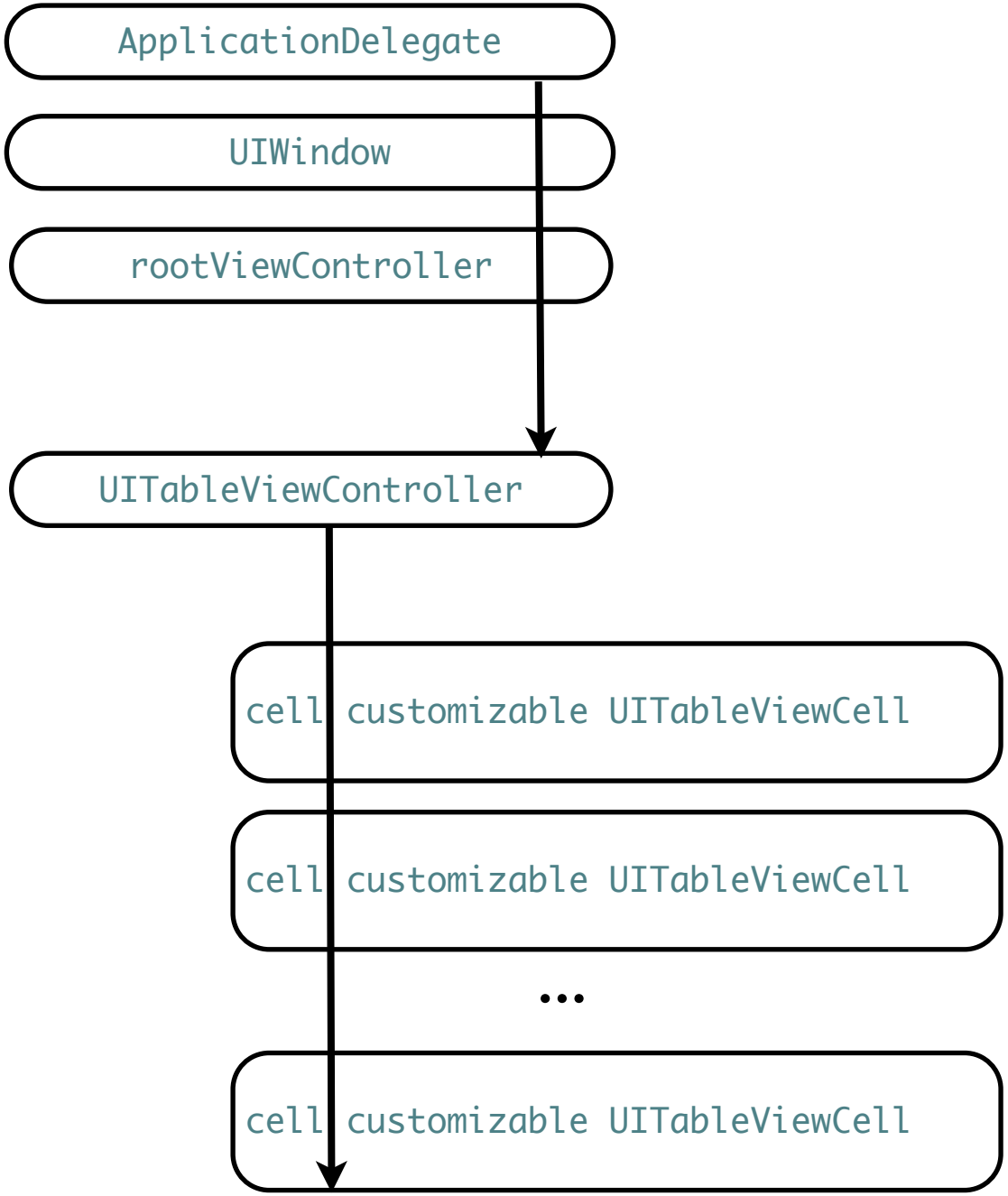
```
- (NSString *)tableView:(UITableView *)tableView titleForHeaderInSection:
(NSInteger)section
{
    return [NSString stringWithFormat:@"%i",section];
}

- (NSArray *)sectionIndexTitlesForTableView:(UITableView *)tableView
{
    return [[NSArray alloc] initWithObjects:@"0",@"1",@"2",@"3", nil];
}
```



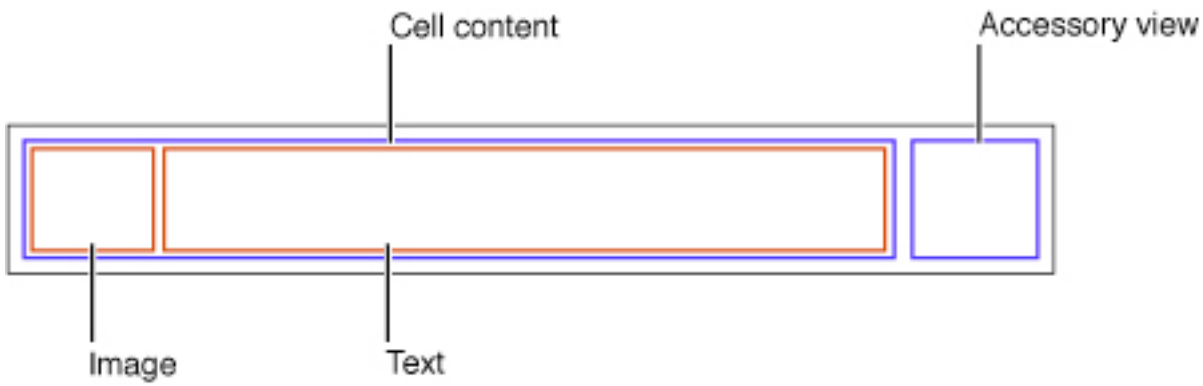
accessory views

```
- (UITableViewCell *)tableView:(UITableView *)tableView
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView
                    dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc]
                    initWithStyle:UITableViewCellStyleDefault
                    reuseIdentifier:CellIdentifier] autorelease];
    }
    // Configure the cell...
   [cell.textLabel setText:
          [NSString stringWithFormat:
          @"%i %i",indexPath.section,indexPath.row]];
    cell.accessoryType = UITableViewCellAccessoryNone;
    return cell;
}
```

```
- (void)tableView:(UITableView *)tableView
    didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    do something...
}
```


ApplicationDelegate
UIWindow
rootViewController
UITableViewController
cell customizable UITableViewCell
cell customizable UITableViewCell
...
cell customizable UITableViewCell

customize UITableViewCell




```
typedef enum {
    UITableViewCellStyleDefault,
    UITableViewCellStyleValue1,
    UITableViewCellStyleValue2,
    UITableViewCellStyleSubtitle
} UITableViewCellStyle;
```

```
[cell.detailTextLabel setText:@"subliminal message"];
cell.imageView.image =[UIImage imageWithContentsOfFile:
    [[NSBundle mainBundle] pathForResource:@"green" ofType:@"png"]];
```

or subclass

- You should add subviews to a cell's content view when your content layout can be specified entirely with the appropriate autoresizing settings and when you don't need to modify the default behavior of the cell.
- You should create a custom subclass when your content requires custom layout code or when you need to change the default behavior of the cell, such as in response to editing mode.