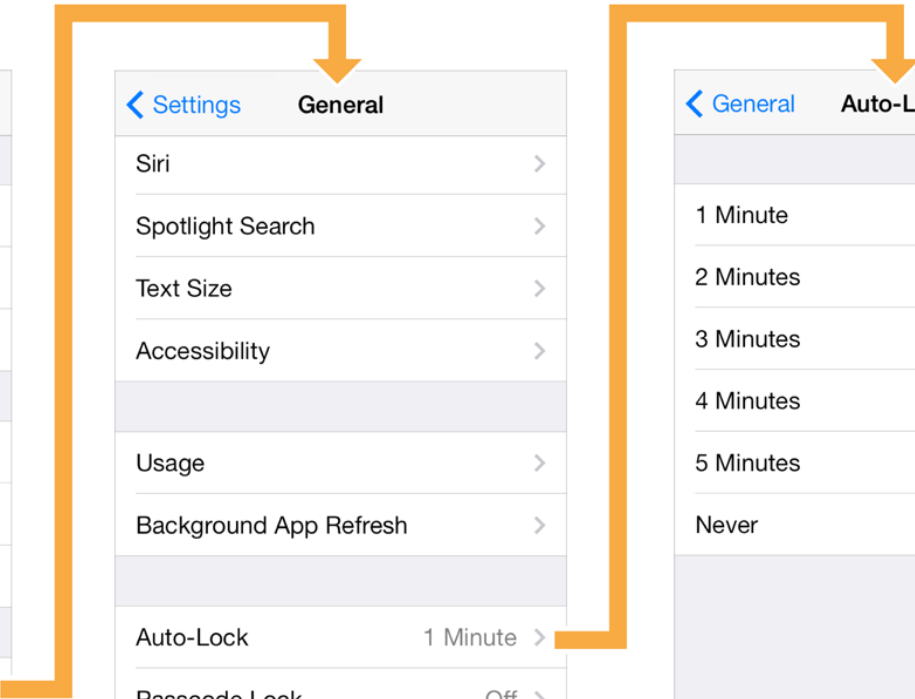
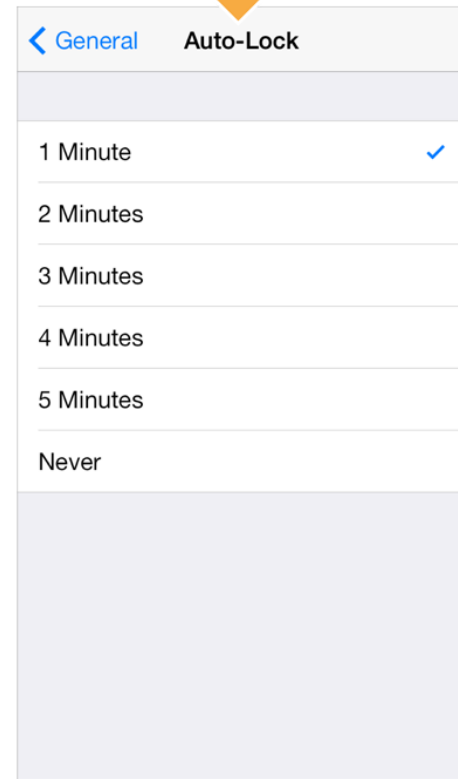
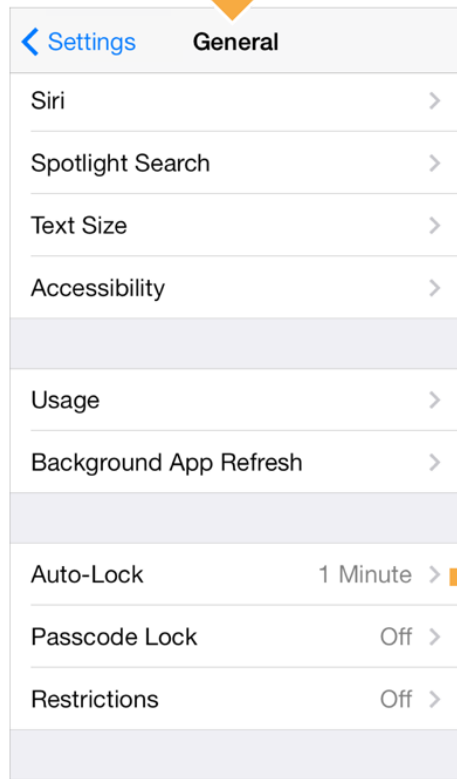
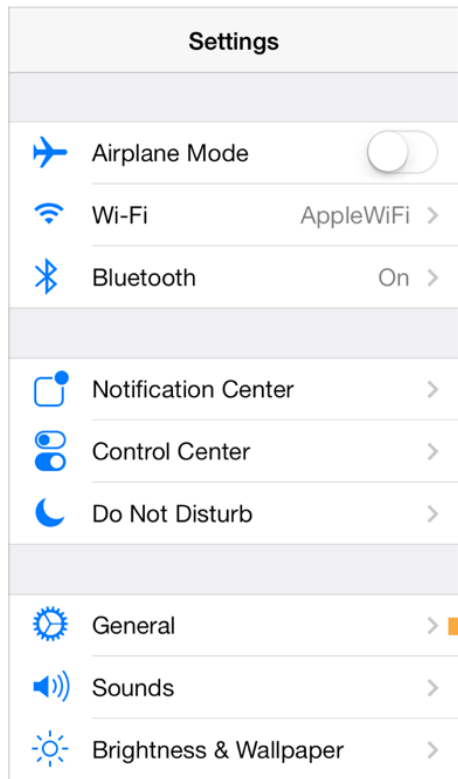


Table View and Navigation Controller

Mar. 24 '16

Navigation Controller



UINavigationController

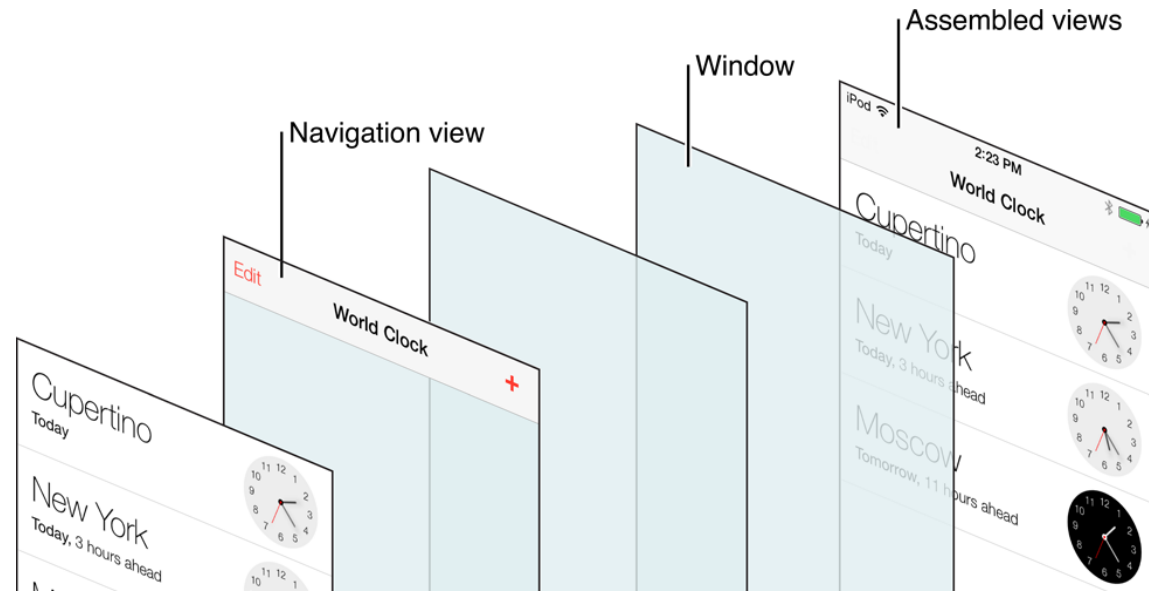
A navigation controller manages a stack of view controllers to provide a drill-down interface for hierarchical content.

The top view controller is the one current showing. You push a view controller in to send users to next level, and pop the top view controller to bring users back.

UINavigationController is one of container view controllers in the iOS and is widely used in the iOS.

For each view controller, use `navigationController` property to access current UINavigationController.

Navigation Bar and Item



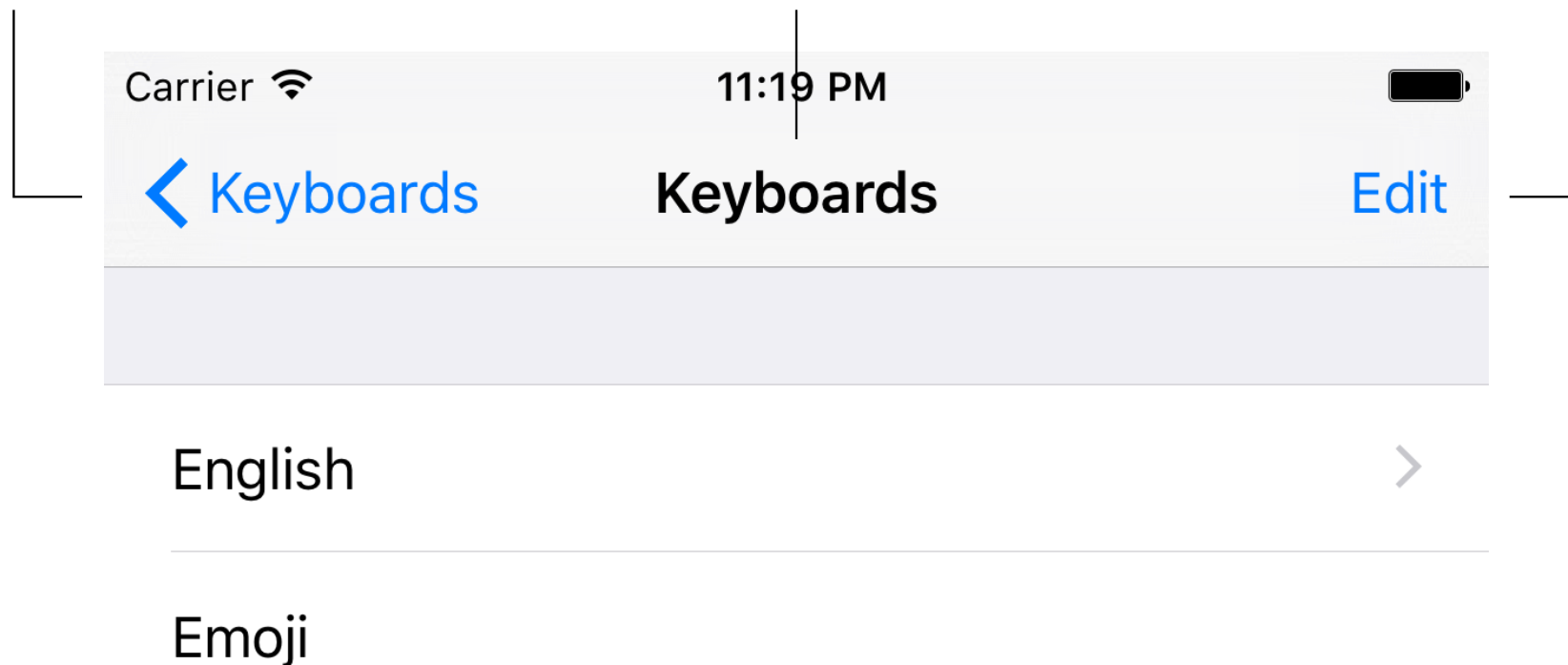
Each view controller has a property `navigationItem` which is used to customize the navigation bar.

UINavigationController

the Left item

the Middle item

the Right item



UINavigationController

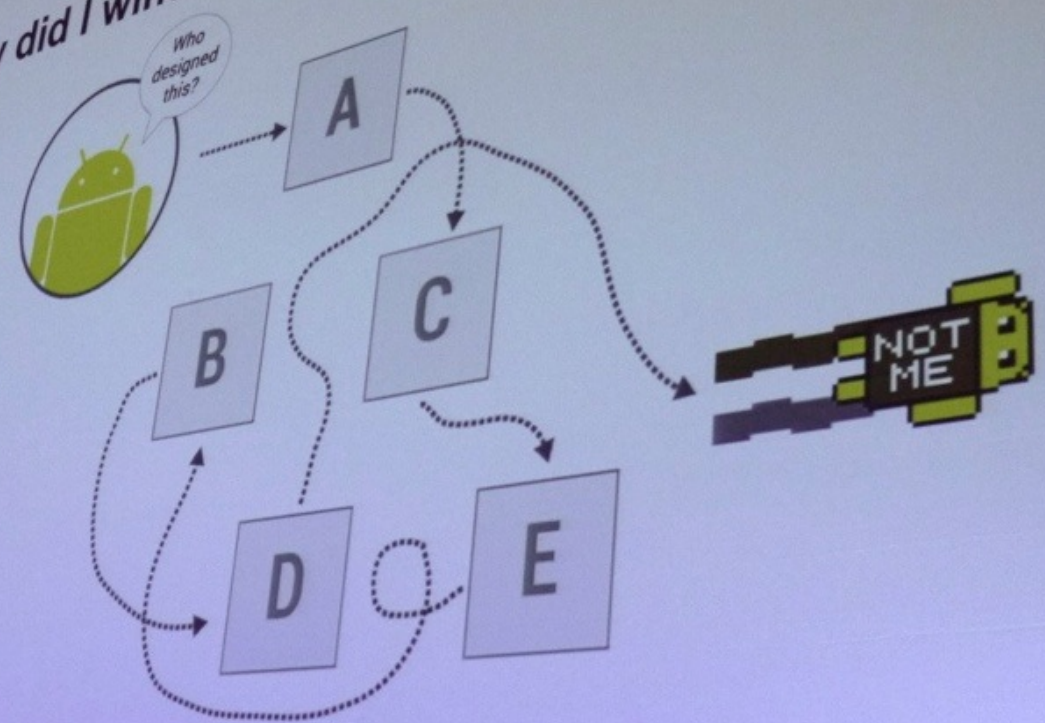
The middle item shows the title of a view controller by default.
Assign a view as titleView of the navigation item to replace it.

The right item is empty by default.

The left item shows a back arrow with title of previous view controller by default.

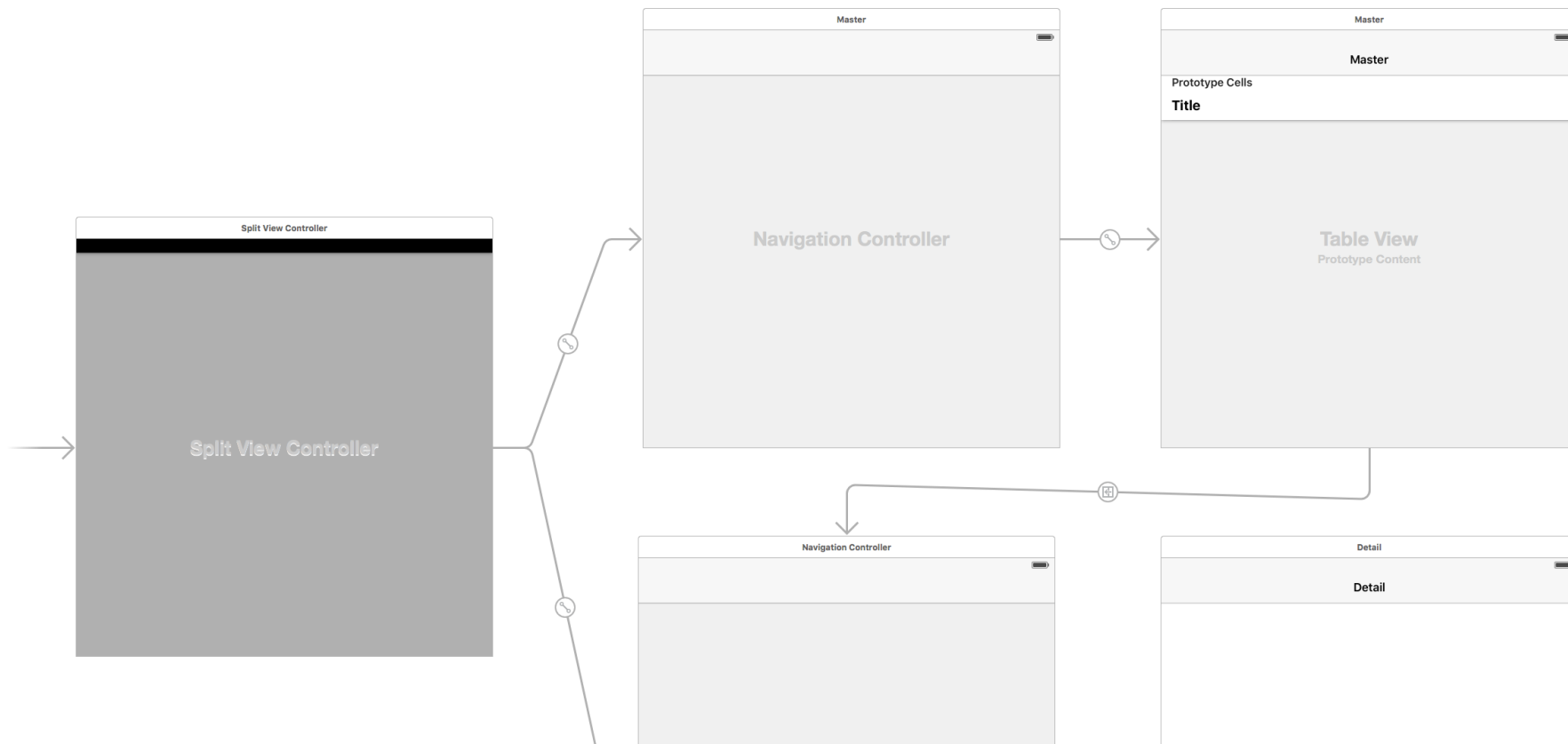
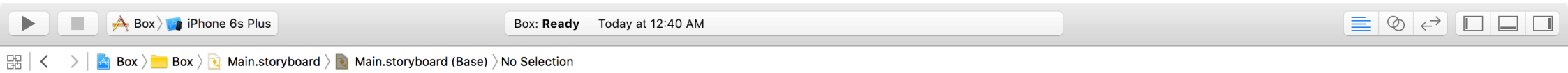
Set leftBarButtonItem or backBarButtonItem of the navigation item to customize.

How did I wind up here?



Storyboard Segue

Storyboard Segue



Storyboard Segue

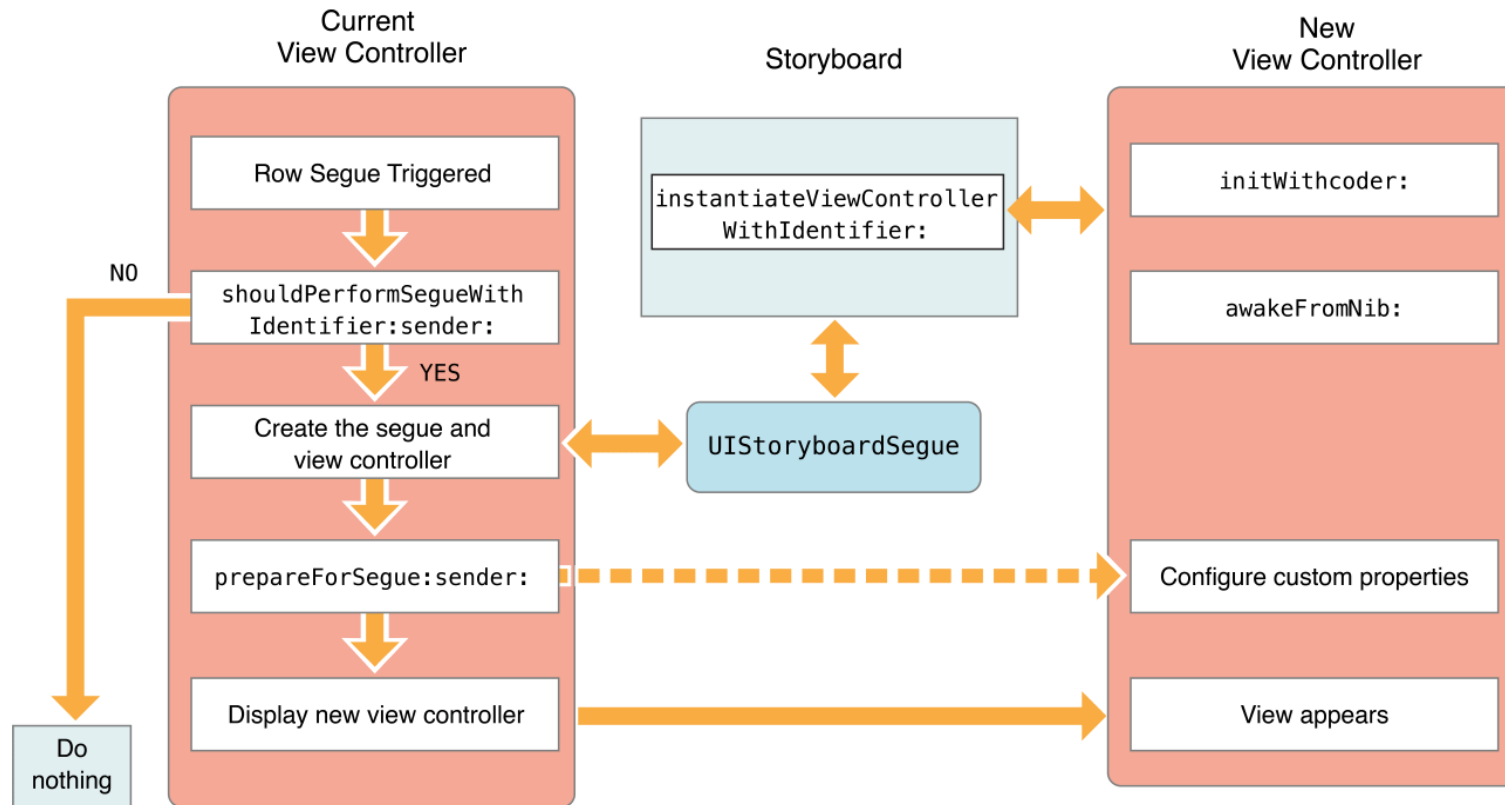
Segue is responsible for performing the visual transition between two view controllers. Segue is also used to represent relationship between view controllers.

Override methods of `UIViewController` to handle segue events.

Use identifier to access segue in code.

Also use *control+drag* to create segues.

Storyboard Segue



Storyboard Segue

Storyboard Segue - *Future Topics*

Common segue patterns

Unwind segue

Custom segues

Perform segue via code

Table View

Table View

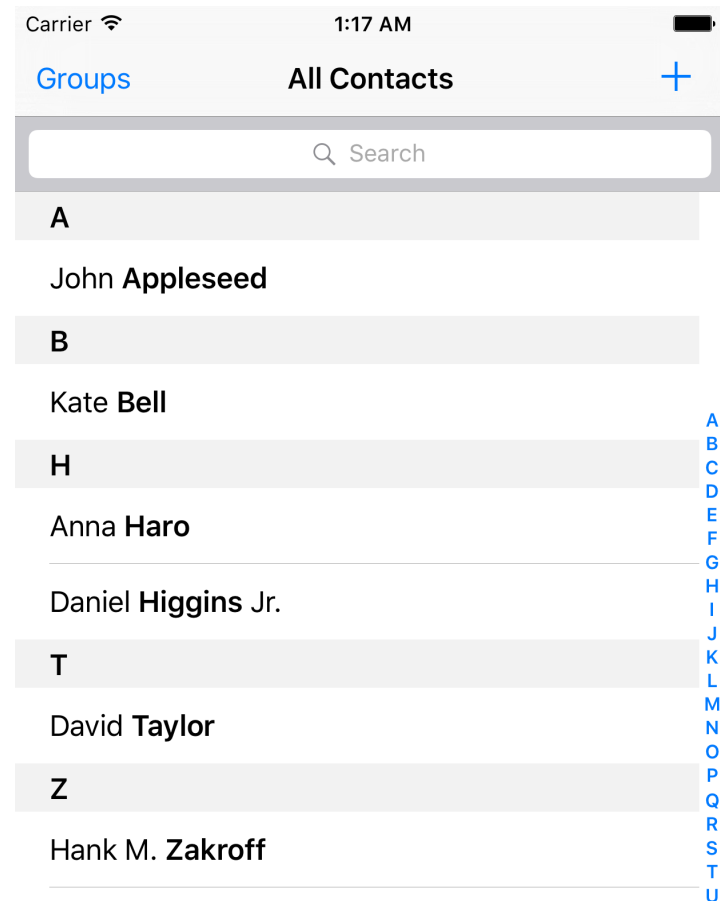
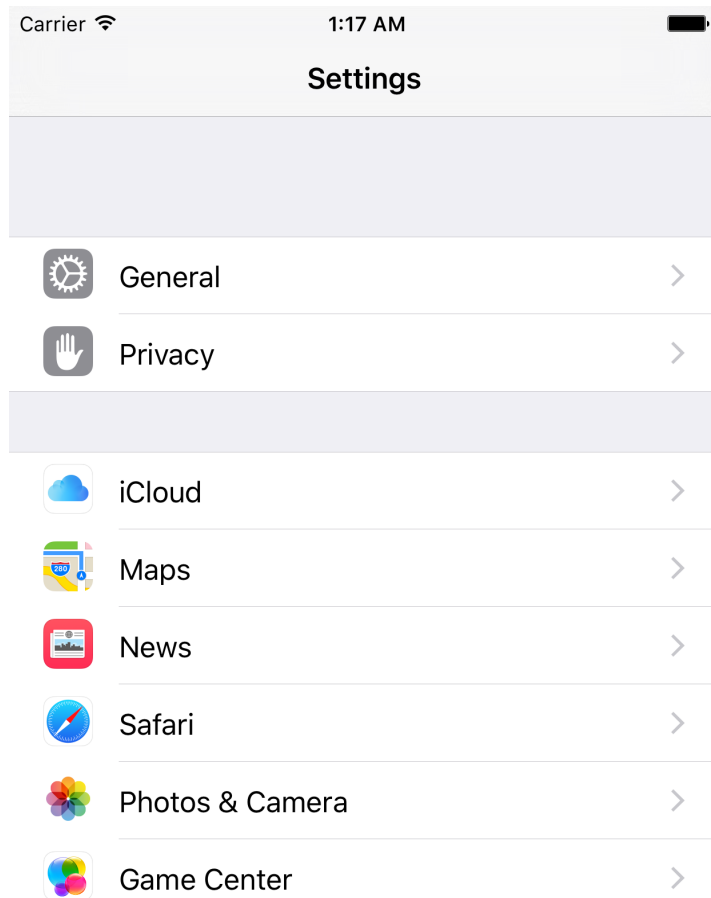
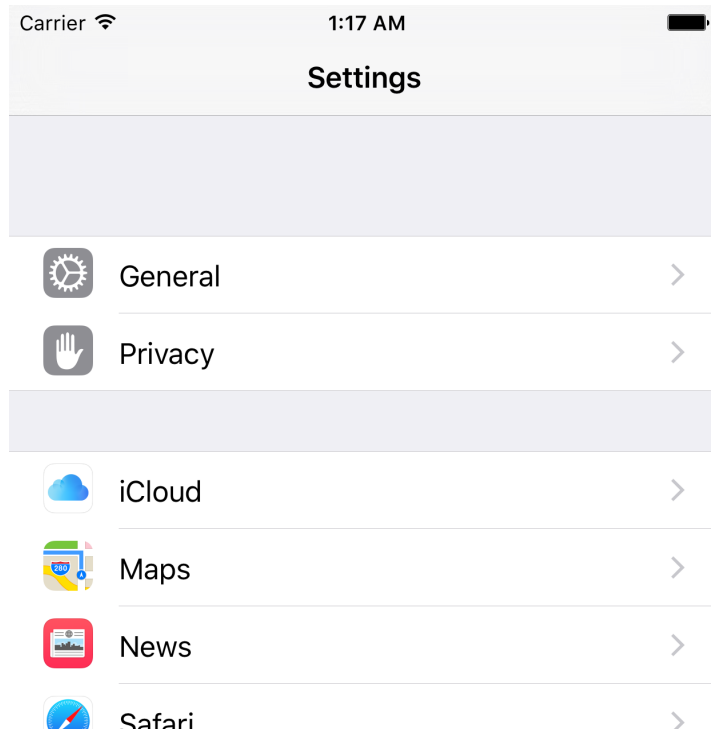


Table View

Grouped Style



Plain Style

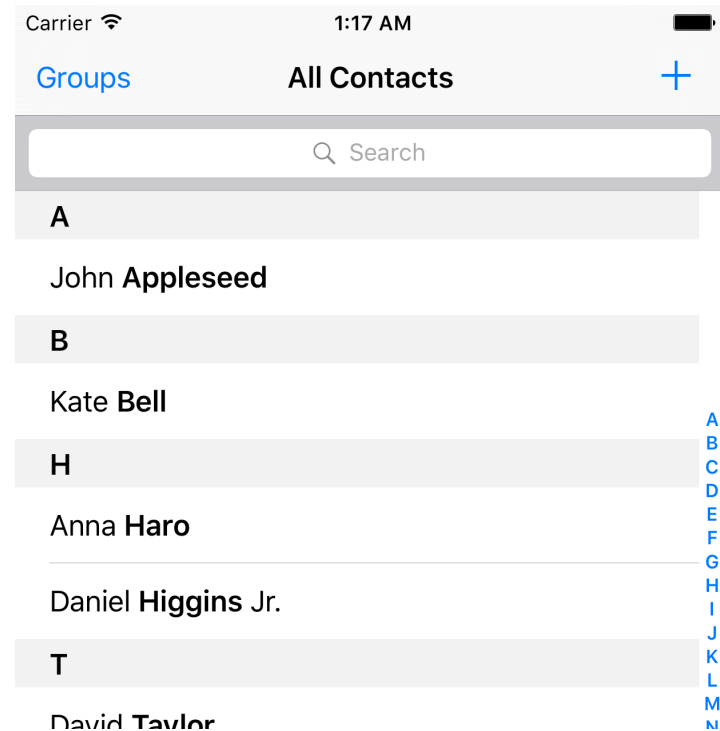


Table View

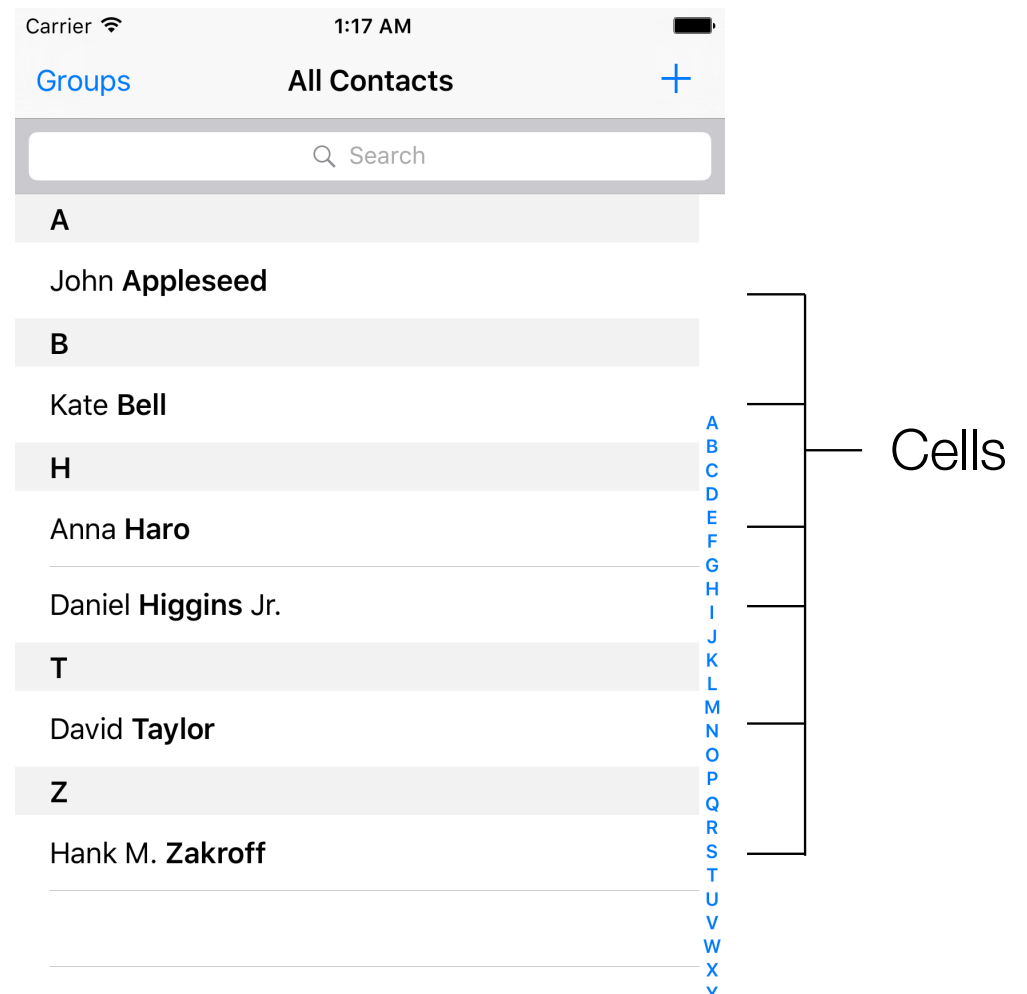
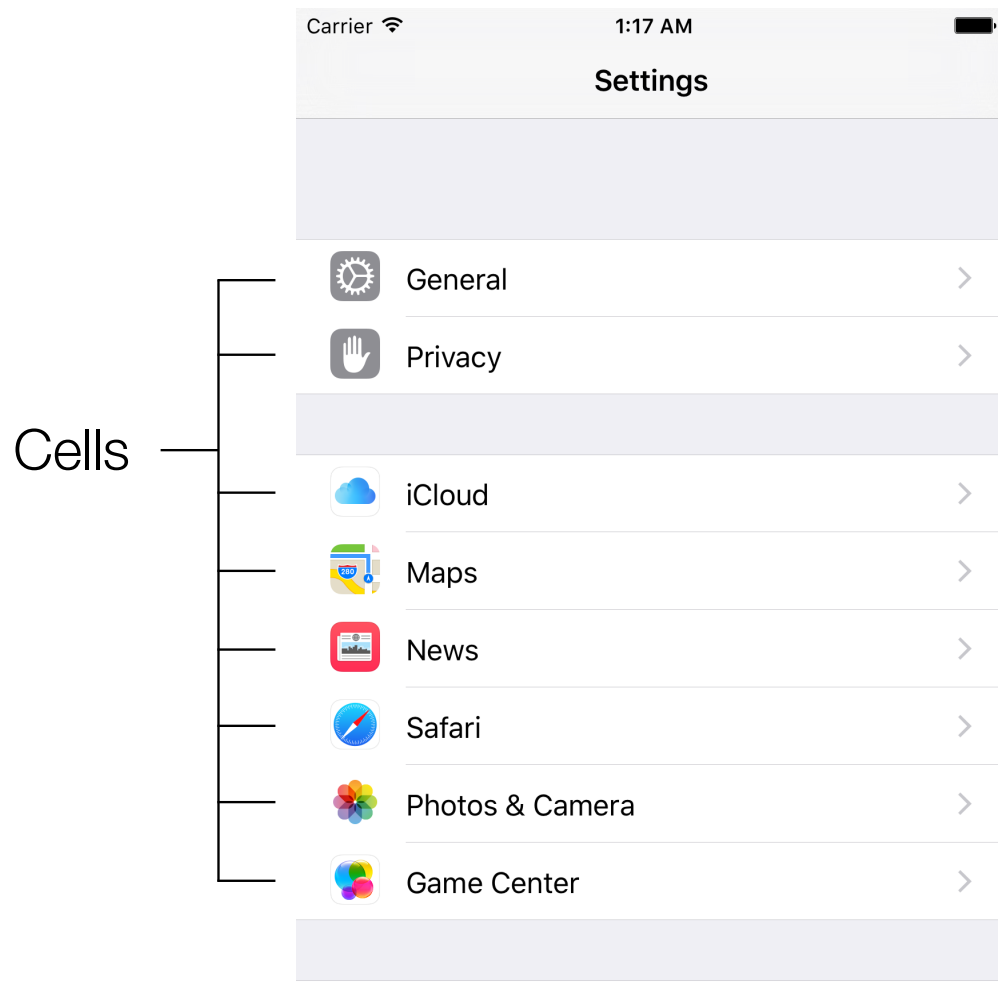


Table View

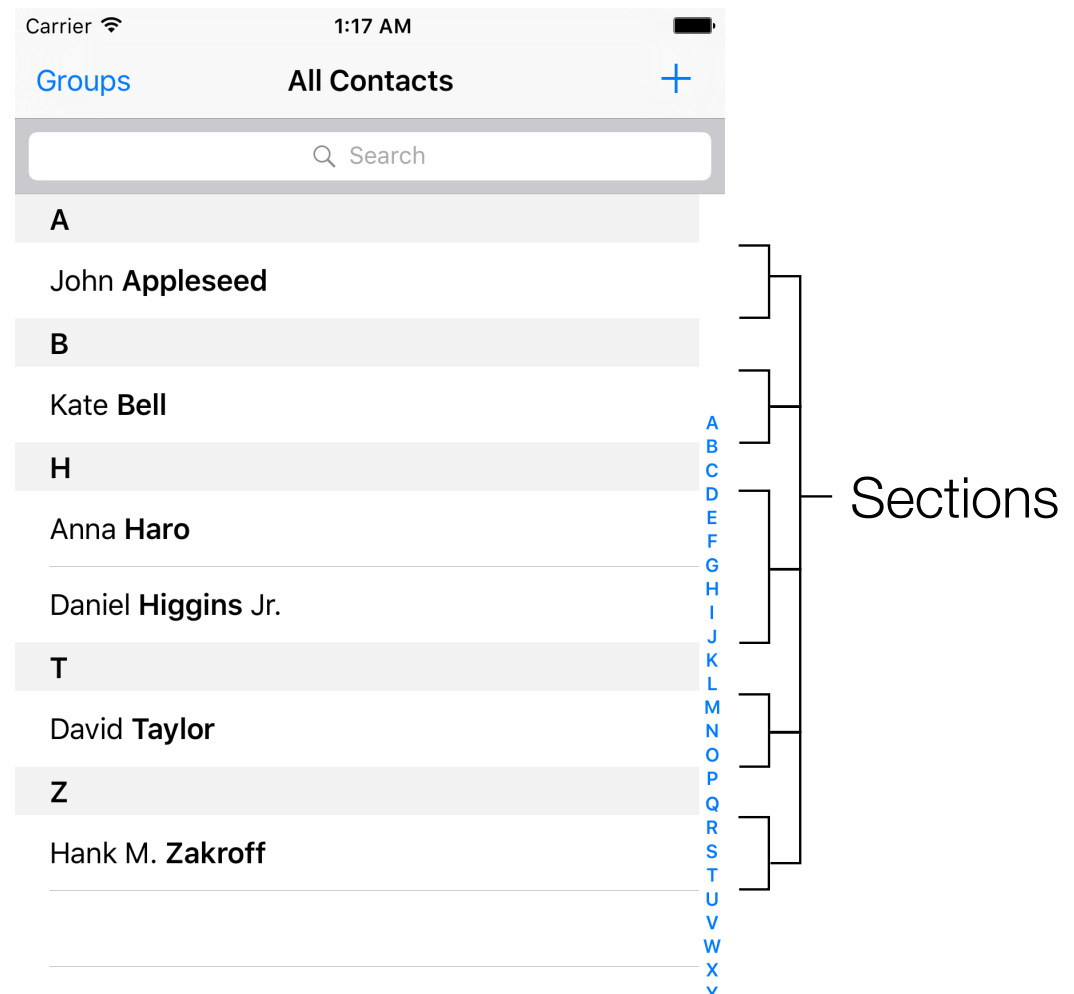
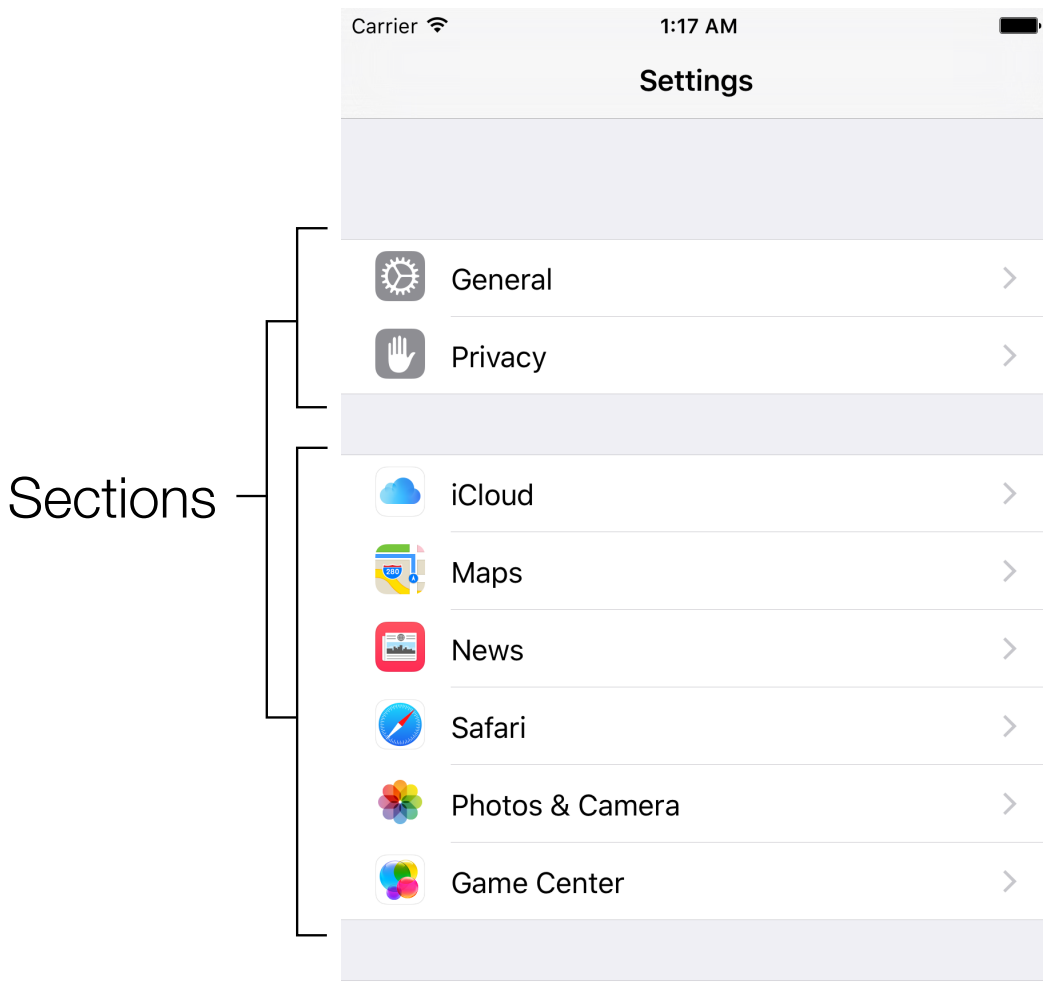
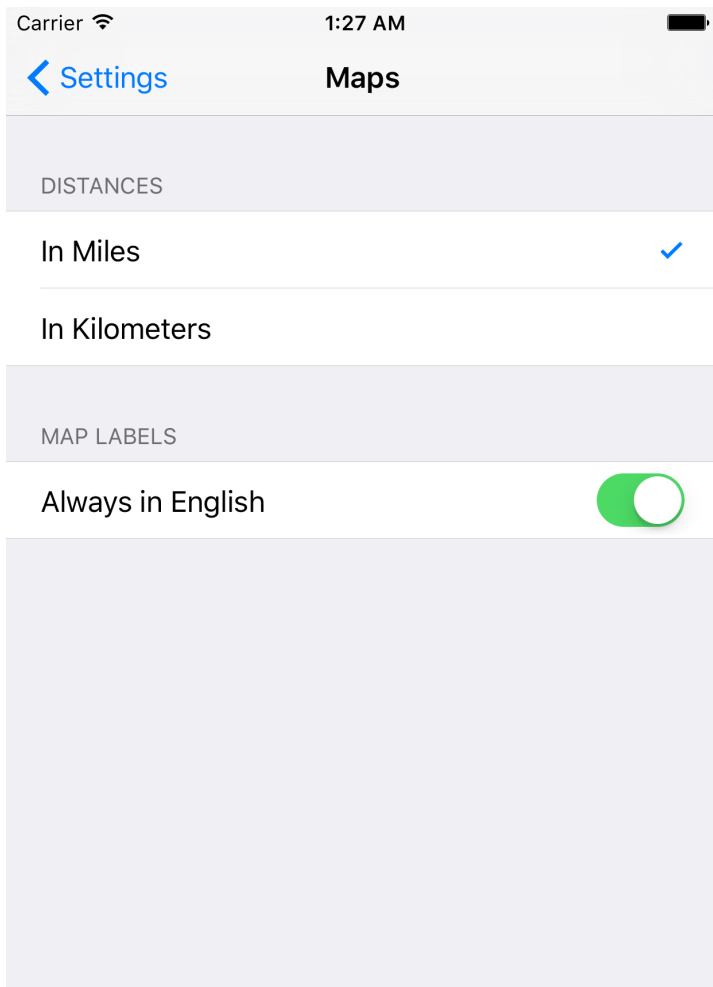
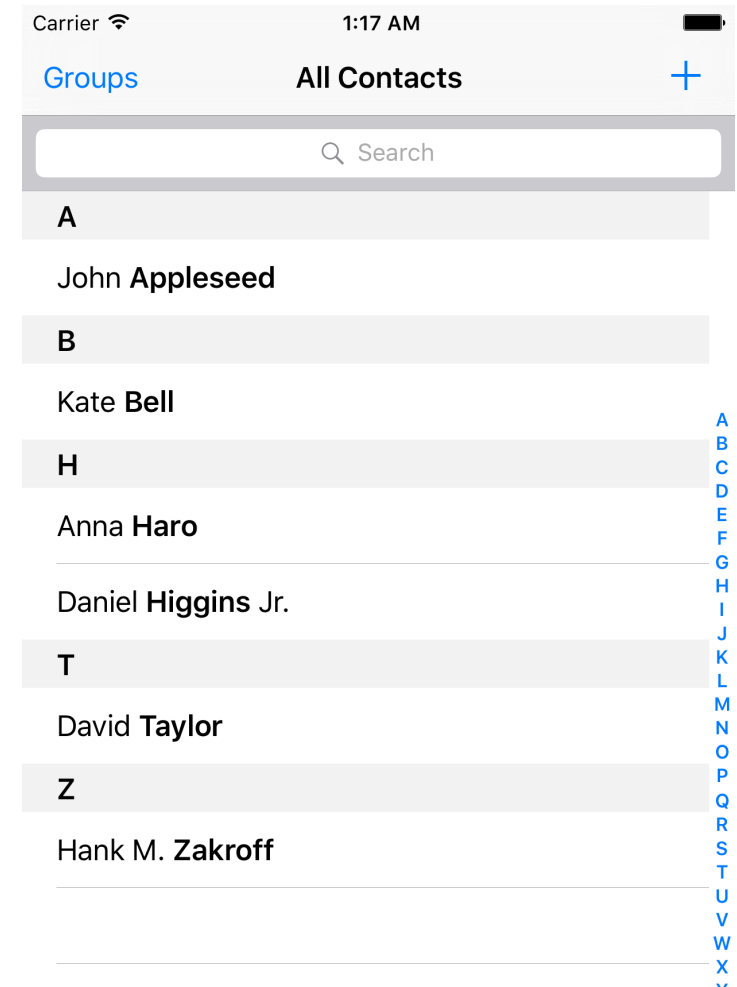


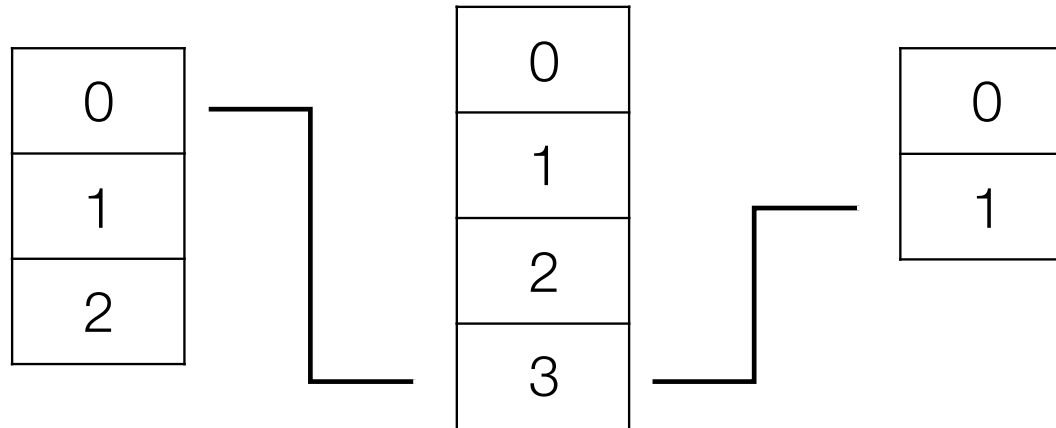
Table View



Section Header





















NSIndexPath



The `NSIndexPath` class represents the path to a specific node in a tree of nested array collections.

`UITableView` uses `NSIndexPath` to represent cell position by section and row.

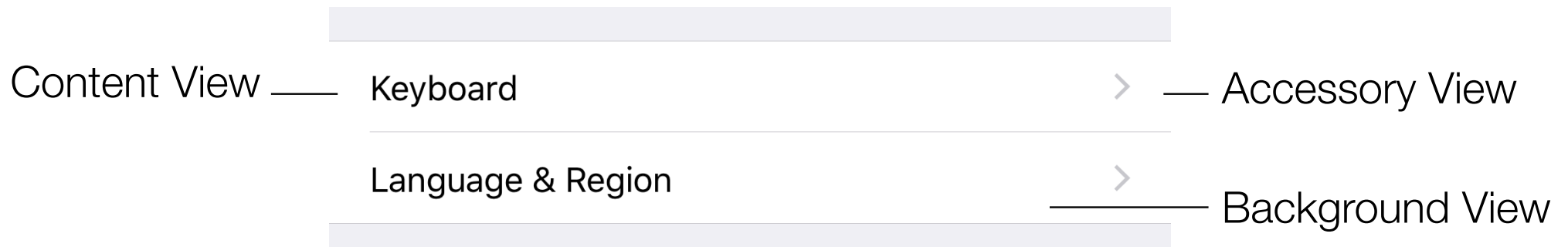
Styles of UITableViewCell

Default Cell Style	Subtitle Cell Style	Value 1 Cell Style	Value 2 Cell Style
 Text Label	 Text Label Detail text label	 Text Label Detail text label	Text Label Detail text label
 Dahlia	 Dahlia This is a dahlia	 Dahlia This is a dahlia	Dahlia This is a dahlia
 Daisies	 Daisies These are daisies	 Daisies These are daisies	Daisies These are daisies
 Dandelion	 Dandelion This is a dandelion	 Dandelion This is a dandelion	Dandelion This is a dandelion
 Echinacea	 Echinacea This is echinacea	 Echinacea This is echinacea	Echinacea This is echinacea
 Lavender	 Lavender This is a field of lavender	 Lavender This is a field of lavender	Lavender This is a field of lavender

UITableViewCell has 4 standard styles.

Use properties like `textLabel` to access these standard style content.

UITableViewCell

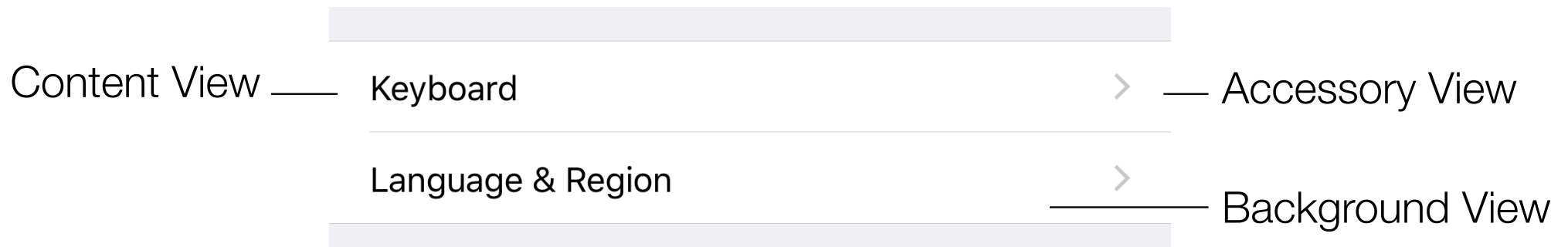


Use tag to access subviews in cells is easier than using outlets.

To use outlets, you have to create a subclass of UITableViewCell.

Storyboard supports to create static UITableViewCell.

UITableViewCell



Select Segue are triggered by events on the cell itself.

Accessory Action (*Segue*) are triggered by the Accessory view.

Table View Performance

Reuse cells. (*UITableView has provided reuse mechanism.*)

Object allocation has a performance cost, especially if the allocation has to happen repeatedly over a short period.

Use opaque subviews and avoid to relayout/redraw of content.

Use static or rendered images, make things easier while reusing. When customizing table view cells, make the subviews of the cell not transparent.

Leave main thread doing UI job.

Fetch resources and perform I/O in other thread. Use preloading and caching.

Delegation Pattern

The `UITableView` uses delegation pattern to fetch data and configure appearance and behavior.

The `UITableViewDataSource` is designed for providing data for the table view. And the `UITableViewDelegate` is used to configure the table view and its events.

The `UITableViewController` is a shortcut which conforms to both the 2 protocols.

Property List

Property List

A file presentation to store Foundation data types.

2 formats: XML and Binary.

OS X and iOS uses Property List to save settings and preferences.

Apps use Info.plist to save app info.

Available data types are: NSArray, NSDictionary, NSString, NSData, NSDate, NSNumber.

“NSDate” is date-time representation in Objective-C, and “NSData” is a wrapper of binary bytes.

Property List

```
let array = [1, 2, 3]
(array as NSArray).writeToFile(path, atomically: true)
let array2: NSArray? = NSArray(contentsOfFile: path)
```

For NSString and NSData, they are written into plain files directly. String is written as text file, and data as binary file.

For NSArray and NSDictionary, they are saved into plist files.

Set “atomically” to use auxiliary file when writing files.

References

[UINavigationController references](#)

[Navigation Controllers](#)

View Controller Catalog for iOS

[Using Segues](#)

View Controller Programming Guide for iOS

[Table View Programming guide for iOS](#)

[Property List Programming Guide](#)

