

iOS @ Taipei - 2016/04/19

UITableView

Nick Lin

UITableView是什麼，可以吃嗎？

- 它是一個展示大量列表數據的UI元件
- 它是一個可直向滾動的的UI元件
- 它是一個可高度客製化的UI元件
- 它真的不好吃，也不能吃，但是要啃懂它

UITableView

UITableView 長什麼樣子

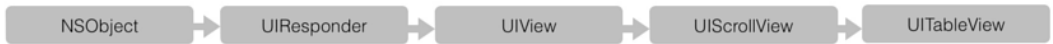


UITableView 從哪裡來 - 1 / 5



- NSObject
is the root class of most Objective-C class hierarchies

UITableView 從哪裡來 - 2 / 5



- UIResponder class defines an interface for objects that respond to and handle events

UITableView 從哪裡來 - 3 / 5



- UIView class
defines a rectangular area on the
screen and the interfaces for
managing the content in that area

UITableView 從哪裡來 - 4 / 5



- UIScrollView class provides support for displaying content that is larger than the size of the application's window

UITableView 從哪裡來 - 5 / 5



- An instance of UITableView (or simply, a table view) is a means for displaying and editing hierarchical lists of information

UITableView

- UITableView的組成

- Table HeaderView
- Table FooterView
- Table Section

Section HeaderView

UITableViewCell

Separator Line

Section FooterView

< Main UITableView 的組成

Table HeaderView

Section 0 HeaderView

Cell for Section 0 , Row 0

Cell for Section 0 , Row 1

Section 0 FooterView

Section 1 HeaderView

Cell for Section 1 , Row 0

Cell for Section 1 , Row 1

Section 1 FooterView

Table FooterView

UITableView 先弄懂的屬性

- **NSIndexPath**

```
extension NSIndexPath {  
    public convenience init(forRow row: Int, inSection section: Int)  
  
    public var section: Int { get }  
    public var row: Int { get }  
}
```

The NSIndexPath class represents the path to a specific node in a tree of nested array collections. This path is known as an index path.

UITableView的重要屬性

```
public init(frame: CGRect, style: UITableViewStyle)
```

- Style 樣式

```
public enum UITableViewStyle : Int {  
    case Plain // regular table view  
    case Grouped // preferences style table view  
}
```

Style (Grouped , Plain)的不同

- Plain

Section Header 永遠在 螢幕顯示 的Section Cell 的上面

Section Footer 永遠在 螢幕顯示 的Section Cell 的下面

UITableView

Style (Grouped , Plain)的不同

Plain

< Main UITableView 的組成
Table HeaderView
Section 0 HeaderView
Cell for Section 0 , Row 0
Cell for Section 0 , Row 1
Cell for Section 0 , Row 2
Cell for Section 0 , Row 3
Cell for Section 0 , Row 4
Cell for Section 0 , Row 5
Cell for Section 0 , Row 6
Cell for Section 0 , Row 7
Section 0 FooterView

< Main UITableView 的組成
Section 0 HeaderView
Cell for Section 0 , Row 6
Cell for Section 0 , Row 7
Cell for Section 0 , Row 8
Cell for Section 0 , Row 9
Section 0 FooterView
Section 1 HeaderView
Cell for Section 1 , Row 0
Cell for Section 1 , Row 1
Cell for Section 1 , Row 2
Section 1 FooterView

Style (Grouped , Plain)的不同

- Grouped

Section Header 永遠在 Section的第一筆Cell 的上面

Section Footer 永遠在 Section的最末筆Cell 的下面

UITableView

Style (Grouped , Plain)的不同

Grouped

< Main UITableView 的組成
Table HeaderView
Section 0 HeaderView
Cell for Section 0 , Row 0
Cell for Section 0 , Row 1
Cell for Section 0 , Row 2
Cell for Section 0 , Row 3
Cell for Section 0 , Row 4
Cell for Section 0 , Row 5
Cell for Section 0 , Row 6
Cell for Section 0 , Row 7
Cell for Section 0 , Row 8

< Main UITableView 的組成
Cell for Section 0 , Row 5
Cell for Section 0 , Row 6
Cell for Section 0 , Row 7
Cell for Section 0 , Row 8
Cell for Section 0 , Row 9
Section 0 FooterView
Section 1 HeaderView
Cell for Section 1 , Row 0
Cell for Section 1 , Row 1
Cell for Section 1 , Row 2

UITableView 怎麼溝通

- DataSource
 - UITableViewDataSource
- Delegate
 - UITableViewDelegate
 - UIScrollViewDelegate
- Self
 - UITableView
 - UIScrollView

UITableViewDataSource

系統提供11個Func， 兩個一定要有 (iOS 9)

一定要有：
 numberOfRowsInSection
 cellForRowAtIndexPath

```
func numberOfSectionsInTableView(tableView: UITableView) -> Int  
  
func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int  
  
func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell
```

UITableViewDelegate

系統提供37個Func, 全都是可選的(iOS 9)

```
func tableView(tableView: UITableView, willDisplayCell cell: UITableViewCell,  
    forRowAtIndexPath indexPath: NSIndexPath)
```

```
func tableView(tableView: UITableView, heightForRowAtIndexPath indexPath:  
    NSIndexPath) -> CGFloat
```

```
func tableView(tableView: UITableView, heightForHeaderInSection section: Int) ->  
    CGFloat
```

```
func tableView(tableView: UITableView, heightForFooterInSection section: Int) ->  
    CGFloat
```

```
func tableView(tableView: UITableView, didSelectRowAtIndexPath indexPath:  
    NSIndexPath)
```

```
func tableView(tableView: UITableView, editActionsForRowAtIndexPath indexPath:  
    NSIndexPath) -> [UITableViewRowAction]?
```

UITableView

UIScrollViewDelegate

系統提供11個Func，沒有一定要有 (iOS 9)

```
func scrollViewWillBeginDragging(scrollView: UIScrollView)
func scrollViewDidScroll(scrollView: UIScrollView)
func scrollViewDidEndDecelerating(scrollView: UIScrollView)
```

UITableView

本身提供很多方法可使用，列幾個常用的

- scrollToRowAtIndexPath
- editing , selectRowAtIndexPath
- separatorStyle , separatorColor , separatorEffect
- tableViewHeader , tableViewFooter
- reloadData , reloadData()
- **beginUpdates() , endUpdates()**

UITableView

本身提供很多方法可使用，列幾個常用的
Cell 相關

- dequeueReusableCellWithIdentifier
- registerNib , registerClass
- cellForRowAtIndexPath 取得indexPath位置上的Cell
- indexPathForCell 取得cell的所在位置
- visibleCells

UIScrollView

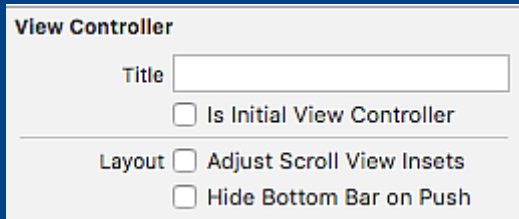
本身提供很多方法可使用，列幾個常用的

- `contentInset`
- `scrollEnabled`
- `scrollRectToVisible`
- `showsHorizontalScrollIndicator`
- `showsVerticalScrollIndicator`
- `indicatorStyle`

UIScrollView

UIScrollView會依據NavigationBar(44px)
跟 System StatusBar(20px)的存在向下移

- `self.automaticallyAdjustsScrollViewInsets = false`



View Controller

Title

☐ Is Initial View Controller

Layout ☐ Adjust Scroll View Insets

☐ Hide Bottom Bar on Push

只要在ViewController
處理，這樣就搞定不會
往下移

iOS @ Taipei

Thank You

Nick Lin