

从 0 开发一款 iOS App



扫码了解课程详情

UIBarButtonItem

UITabBar

UITabBarController

M

V

C

UITabBarItem

UINavigationController

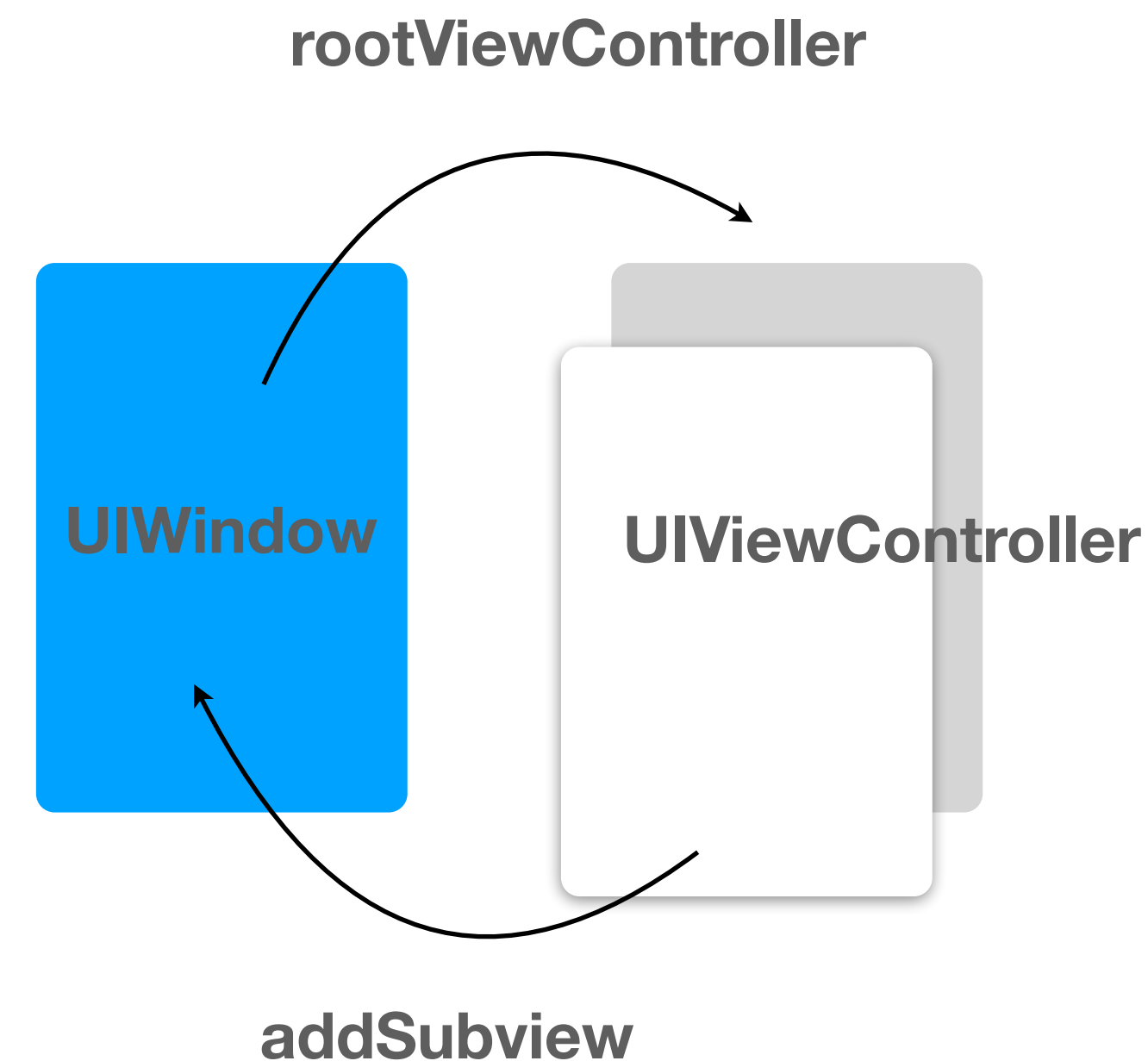
UINavigationController

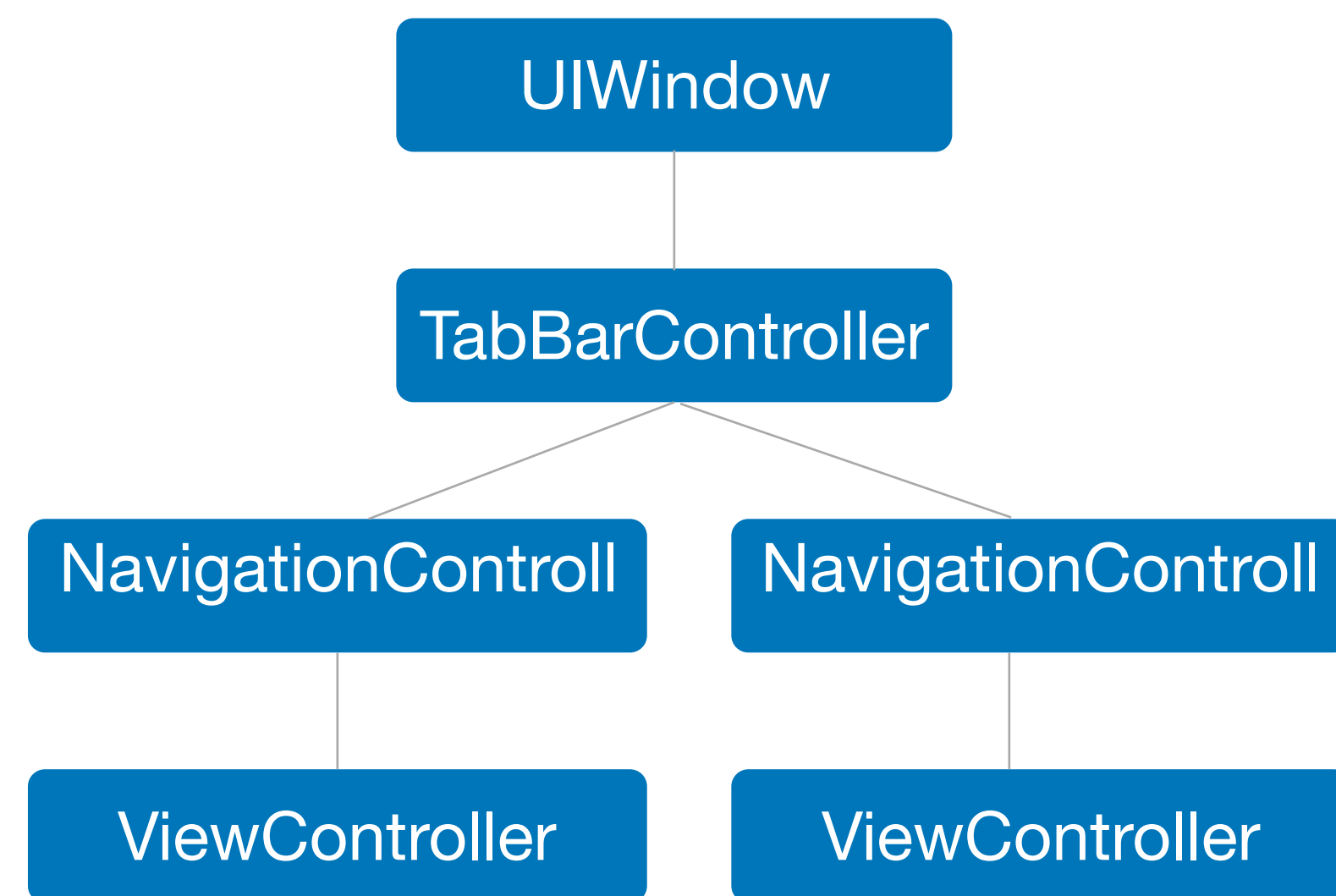
UIWindow

- 特殊形式的 UIView，提供 App 中展示内容的基础窗口
- 只作为容器，和 ViewController 一起协同工作
- 通常屏幕上只存在、展示一个 UIWindow

- 使用 storyboard 会帮我们自动创建
- 手动创建

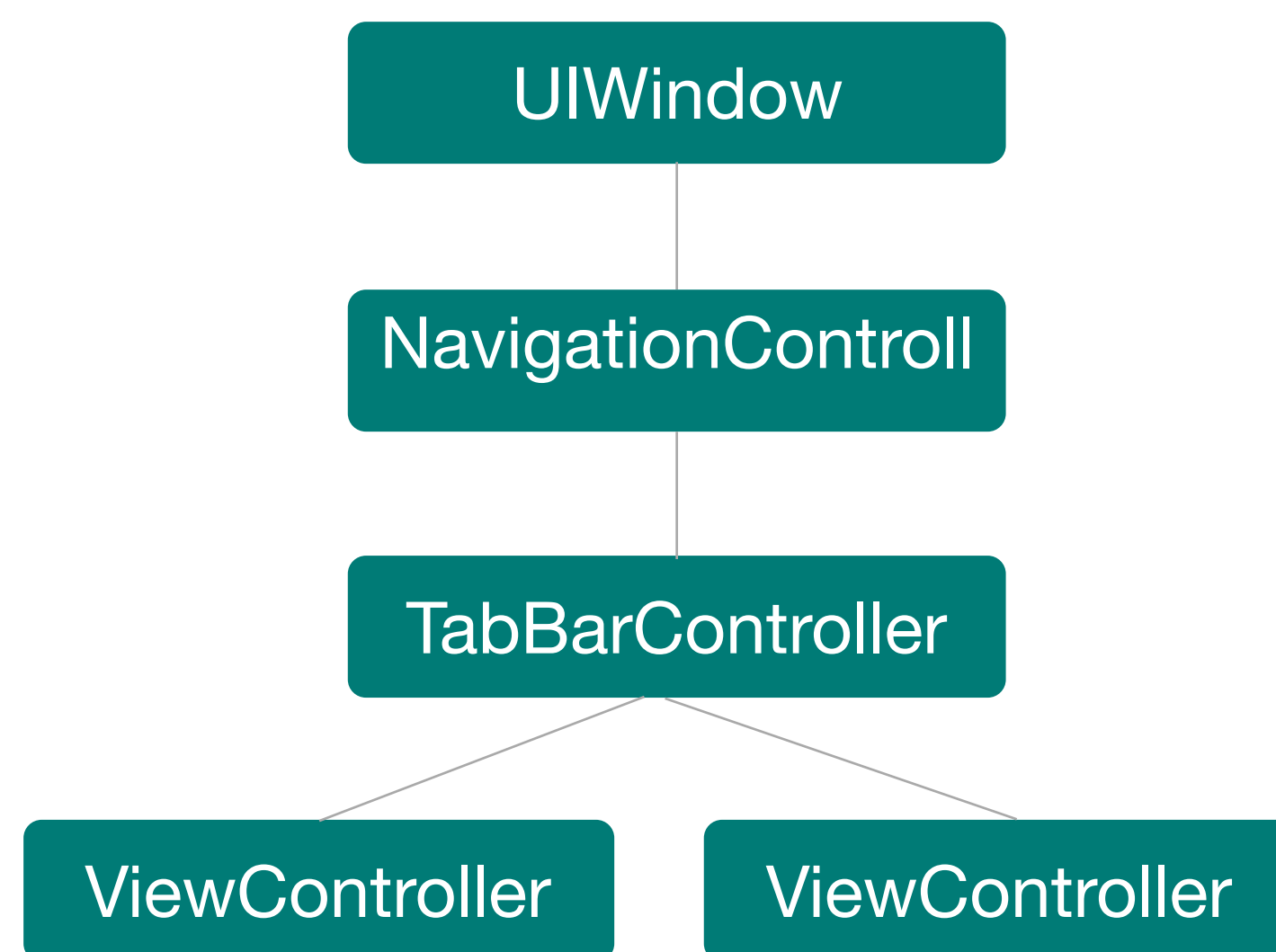
1. 创建 UIWindow
2. 设置 rootViewController
3. makeKeyAndVisible





Apple 推荐



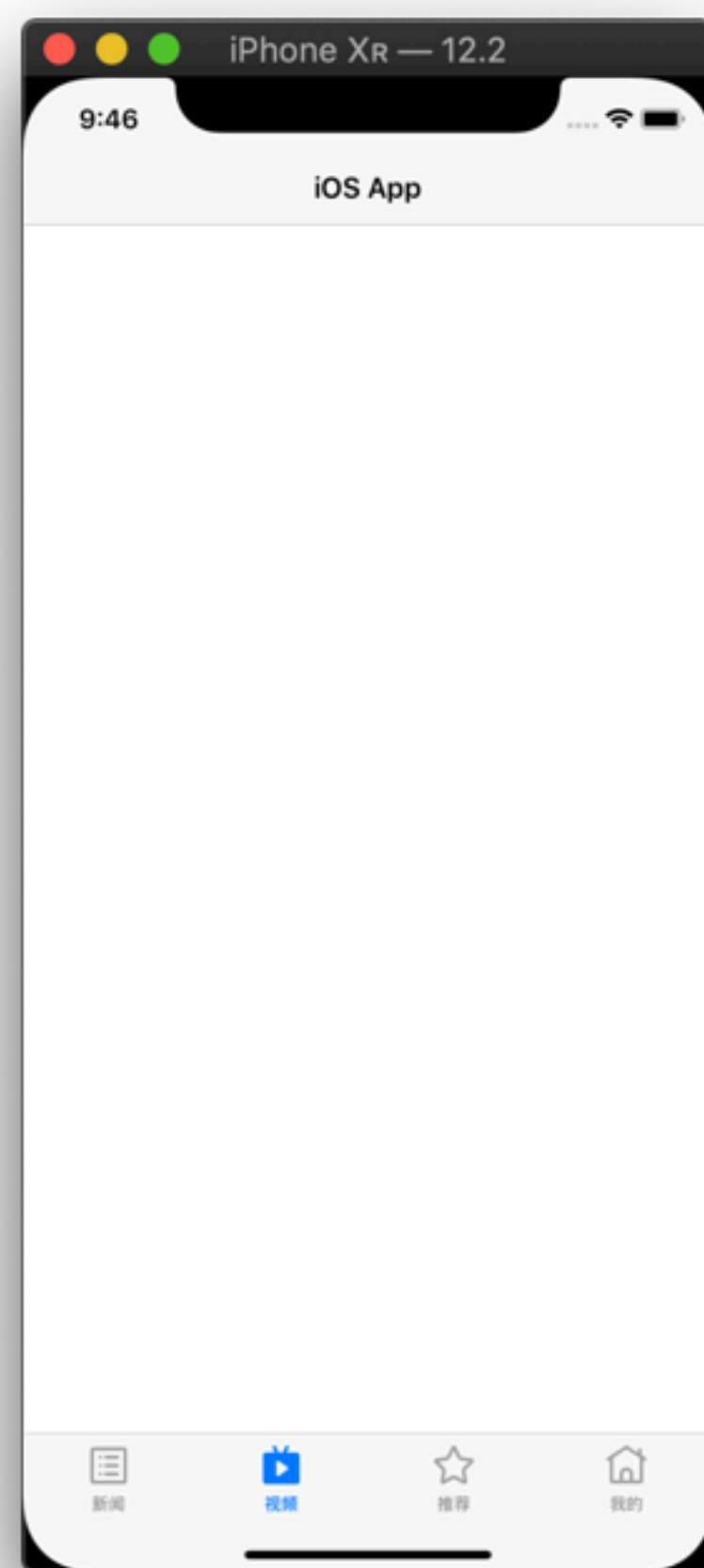


方便、常用



delegate 设计模式

UITabBarController - 自动处理点击之后的页面切换



// 点击处理

```
if (不切换 ViewController) {  
    return;  
}
```

// 切换 ViewController

// 切换完成

(BOOL) tabBarController:
shouldSelectViewController:

(void) tabBarController:
didSelectViewController:

需要执行自定义操作的对象

delegate

协议

delegate 设计模式

1. 设置 self 为 delegate 的接收者，也就是实现方法的对象

```
tabBar.delegate = self
```

2. 根据需求按需实现方法

```
- (BOOL)tabBarController:(UITabBarController *)tabBarController  
shouldSelectViewController:(UIViewController *)viewController{  
    return YES;  
}  
  
- (void)tabBarController:(UITabBarController *)tabBarController  
didSelectViewController:(UIViewController *)viewController{  
    //播放这个viewController的第一个视频  
}
```

UITabBarController 举例

```
@property(nullable, nonatomic,weak) id<UITabBarControllerDelegate> delegate;
```


delegate 设计模式

设计者	使用者
提供一些使用者可以自定义的操作	设置 <code>delegate = self</code>
<code>@optional/@required</code> 注解	按需实现方法
提供 <code>@property - delegate</code>	
在对应的时机，让 <code>delegate</code> 执行对应方法	

列表



- 数据量大
- 样式较为统一
- 通常需要分组
- 垂直滚动
- 通常可视区只有一个



视图的复用



UITableView



UITableViewDataSource

UITableView 作为视图，只负责展示，协助管理，不管理数据

需要开发者为 UITableView 提供展示所需要的数据及 Cell

通过 delegate 的模式，开发者需要实现 UITableViewDataSource

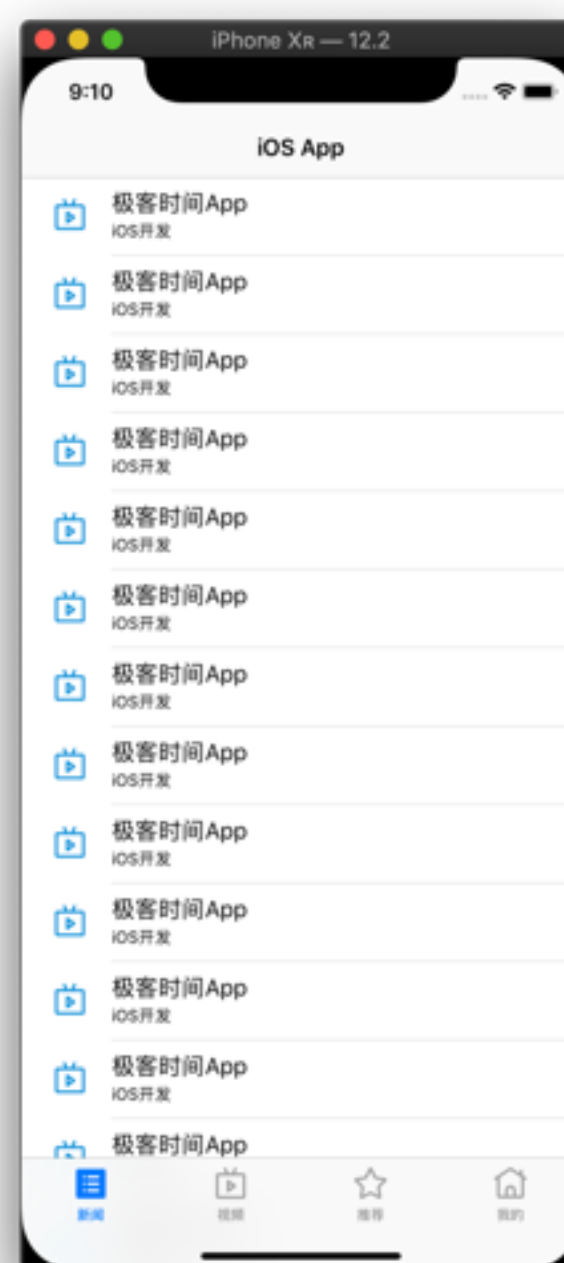
@require

- numberOfRowsInSection:(NSInteger)section;
- cellForRowAtIndexPath:(NSIndexPath *)indexPath;

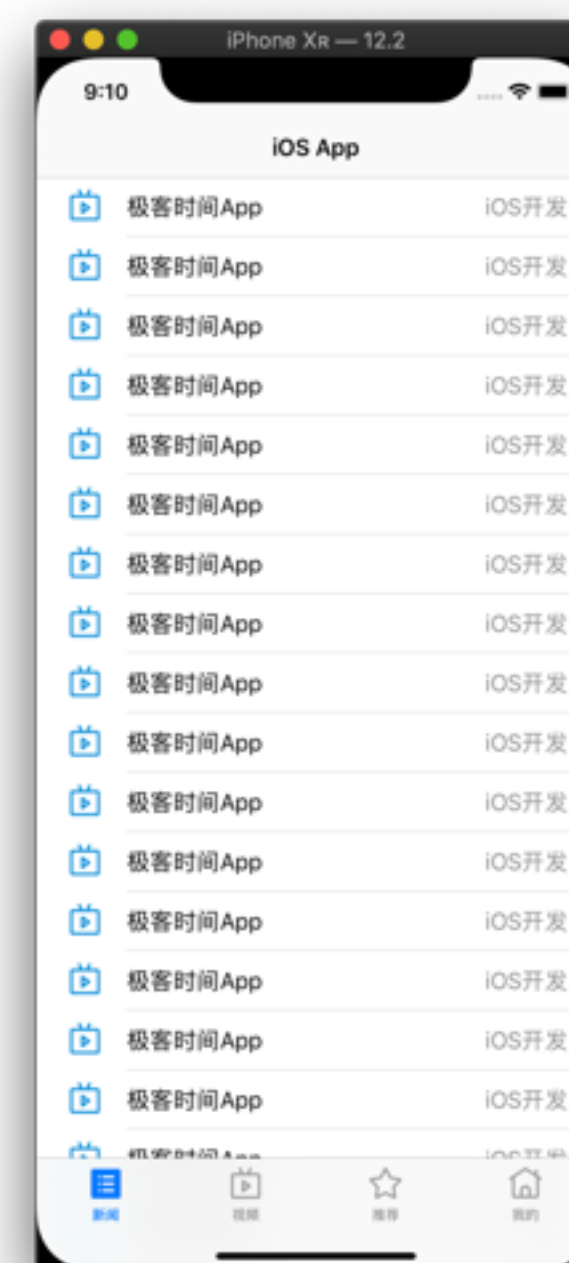
UITableViewCell 默认提供的样式



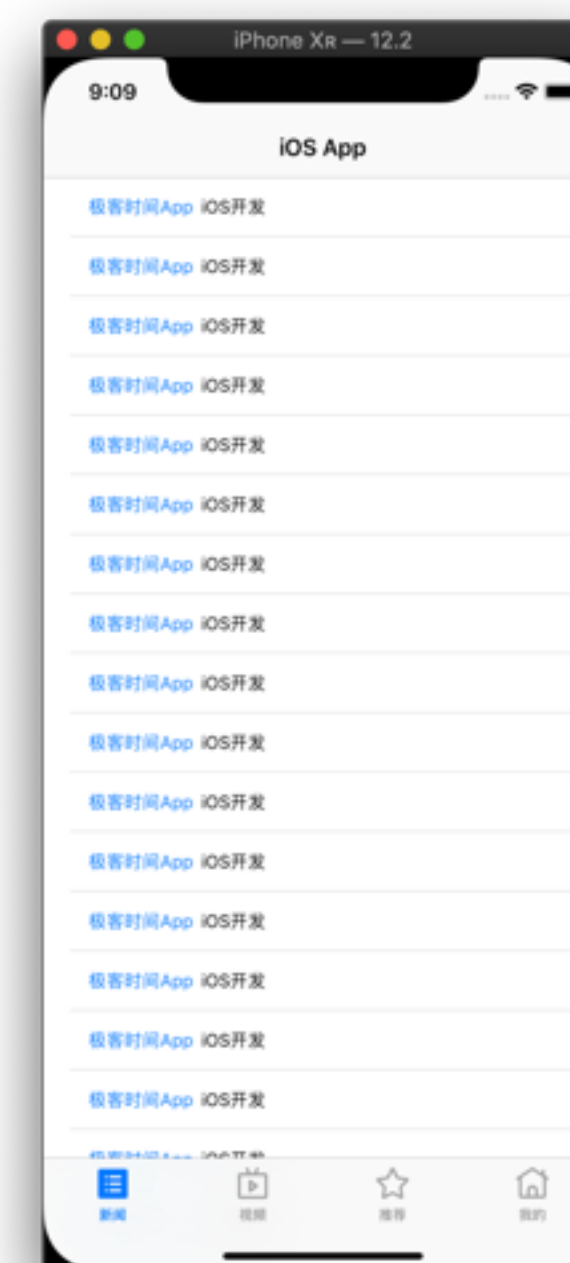
UITableViewCellStyleDefault



UITableViewCellStyleSubtitle

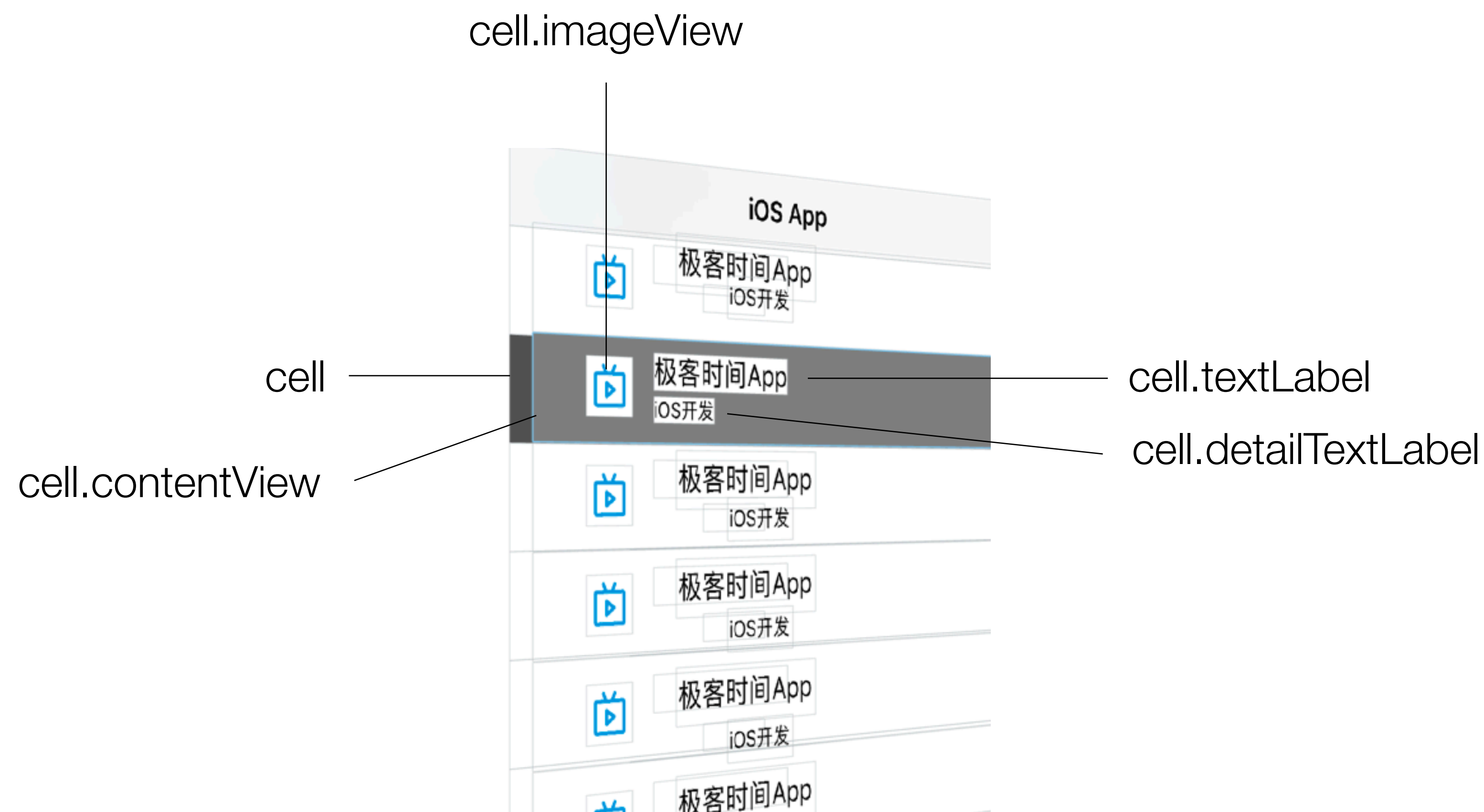


UITableViewCellStyleValue1



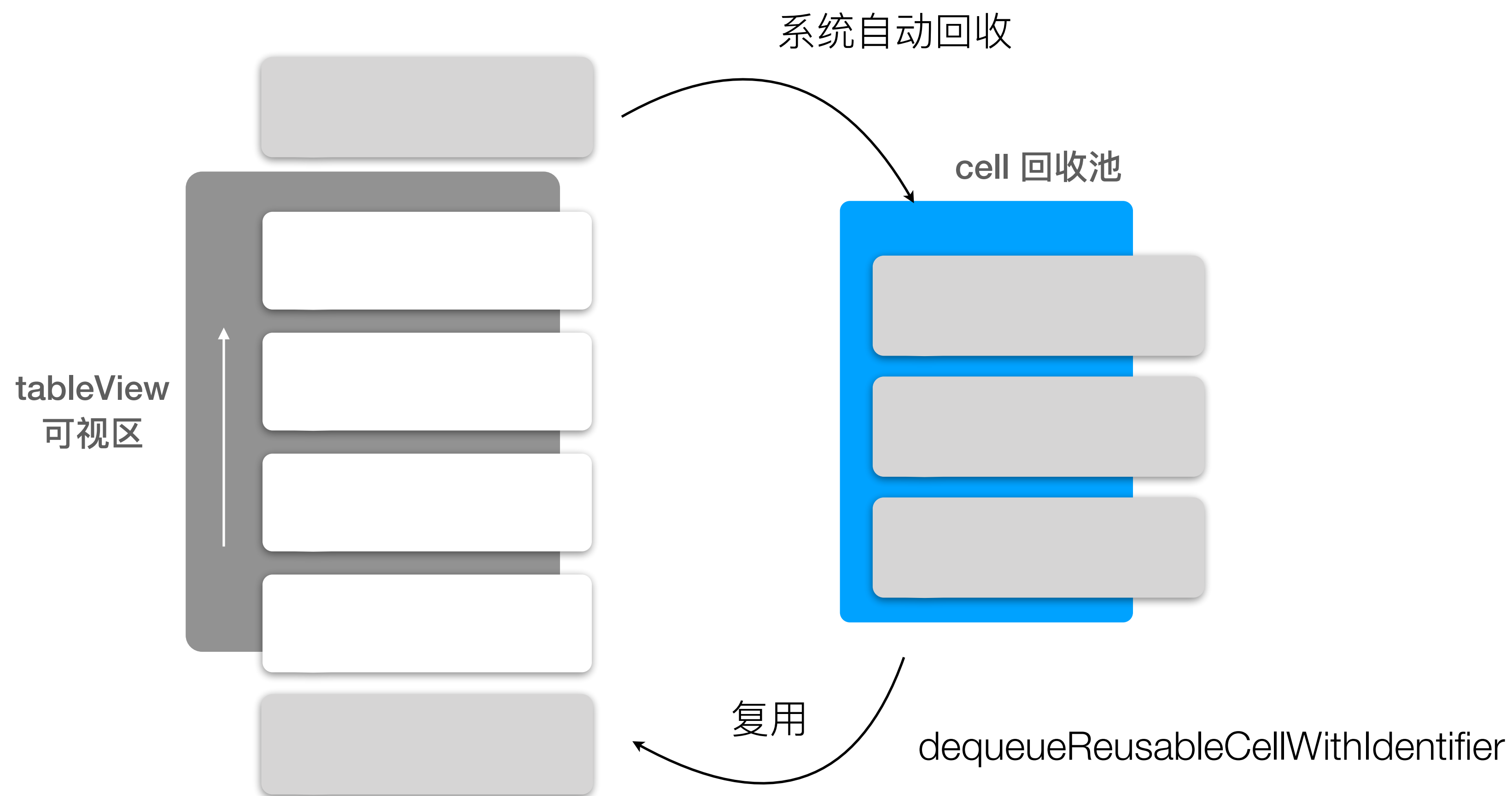
UITableViewCellStyleValue2

UITableViewCell 默认提供展示文字和图片

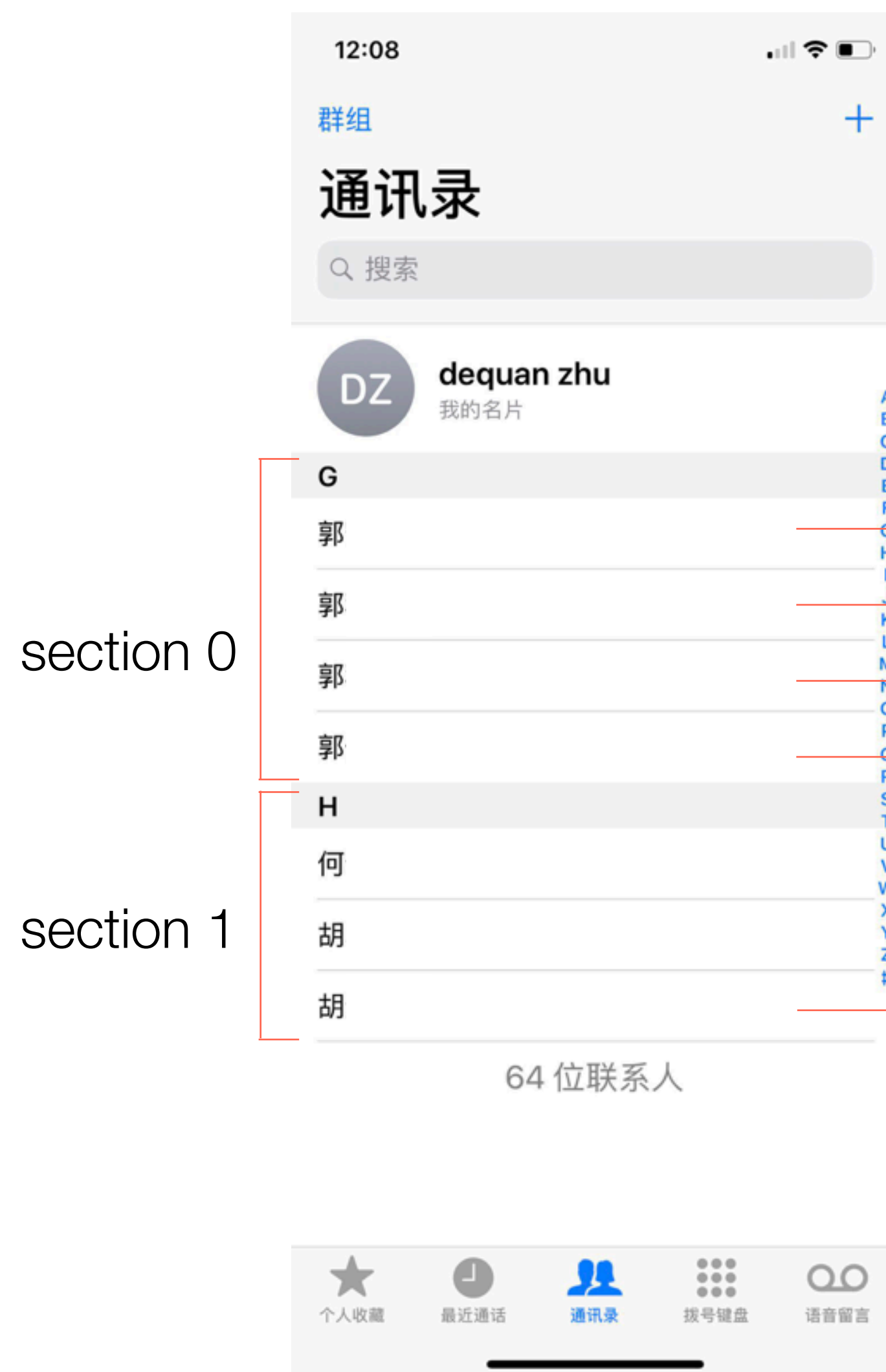


UITableViewCell 的重用

系统提供复用回收池，根据 reuseIdentifier 作为标识



UITableView



NSIndexPath (UIKitAdditions)

indexPath.section = 0

indexPath.row = 0

row 0

row 1

row 2

row 3

indexPath.section = 0

indexPath.row = 3

indexPath.section = 1

indexPath.row = 2

UITableViewDelegate

- 提供滚动过程中，UITableViewCell 的出现、消失时机
- 提供 UITableViewCell 的高度、headers 以及 footers 设置
- 提供 UITableViewCell 各种行为的回调（点击、删除等）

UITableView 基本使用

- 提供最基础的列表类型视图组件
 - 提供默认基础的 UITableViewCell 样式、header 和 footer 的管理
 - 提供针对 UITableViewCell 的复用回收逻辑
 - 提供列表基础功能，如点击、删除、插入等
-
1. 创建 UITableView，设置 delegate 和 datasource，通过两个 delegate
 2. 选择实现 UITableViewDataSource 中方法，行数、cell 复用
 3. 选择实现 UITableViewDelegate 中方法（高度、headerFooter、点击）



扫码了解课程详情