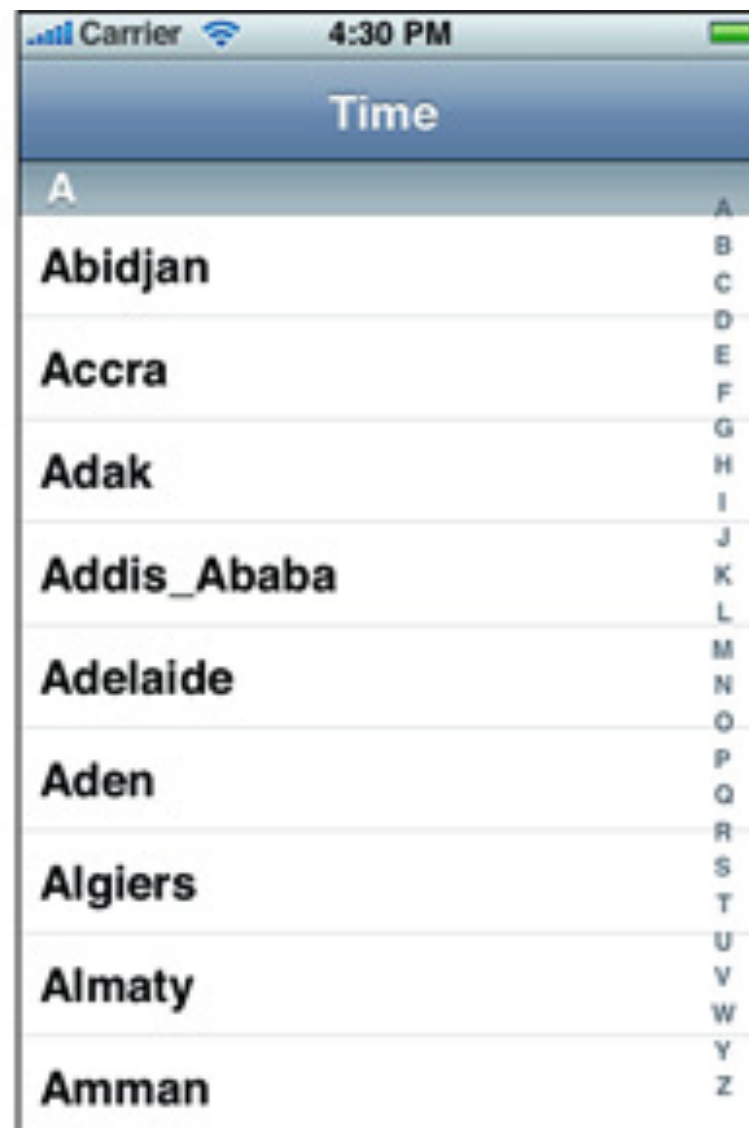
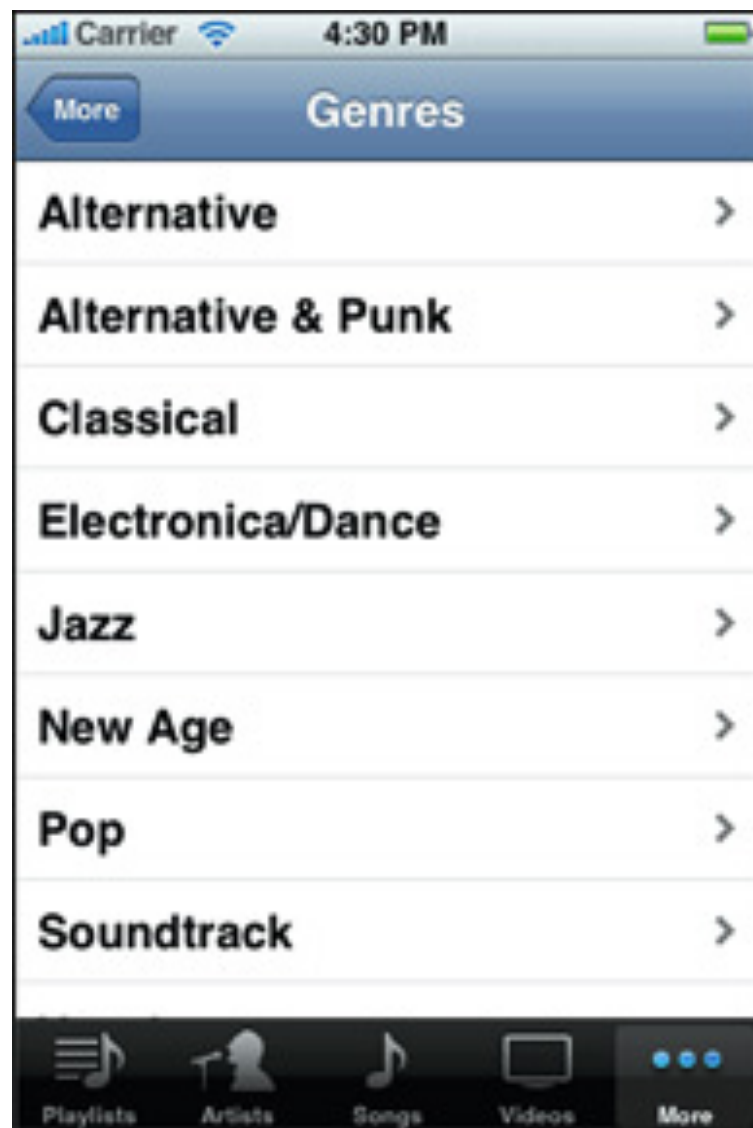


UITableView

UITableView

UIScrollView 를 상속받아 리스트 형태로 반복되는 아이템을 보여주기 위한 뷰

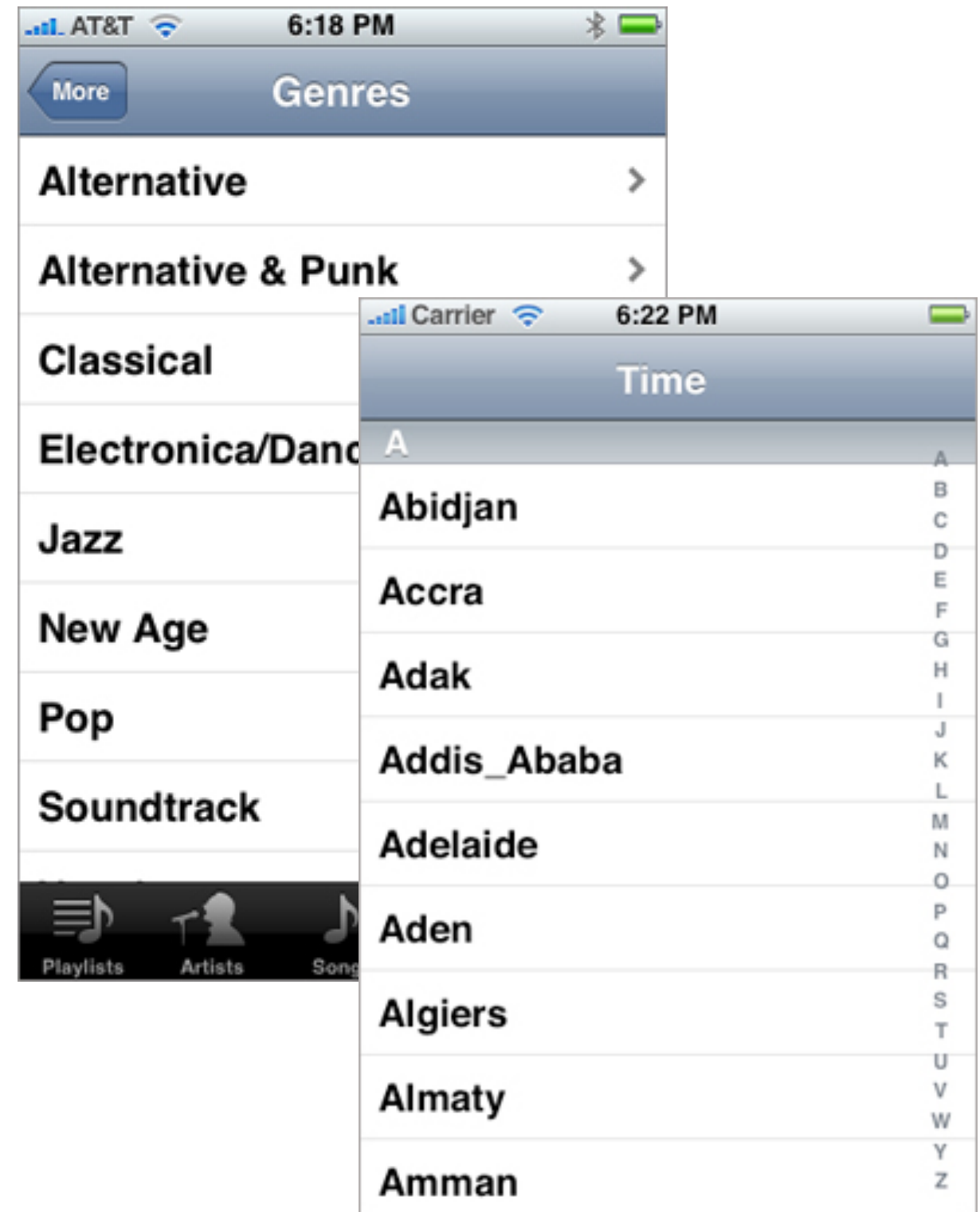
Plain Style / Grouped Style



UITableView

Plain Table Views

- 기본 스타일
- 하나 이상의 섹션을 가질 수 있으며
각 섹션은 다시 하나 이상의 로우를 가짐
- 각 섹션은 그 자신의 헤더/푸터 뷰를
가질 수 있고 해당 섹션이 보여질 때
헤더뷰는 Top, 푸터뷰는 Bottom 에 고정
- 섹션을 빠르게 검색할 수 있도록
Indexed List 를 설정 가능



UITableView

Grouped Table Views

- 관련된 아이템들을 하나의 그룹으로 그룹핑
- 데이터 계층 구조에 대한 상세 정보를 표현하는 데 적합한 형태
- Plain 스타일과 달리 헤더뷰와 푸터뷰가 Top, Bottom 에 고정되지 않음



UITableViewDataSource

TableView 에서 표현할 데이터를 제공하는 데 사용하는 프로토콜

Providing the Table View Data

`dataSource`

The object that acts as the data source of the table view.

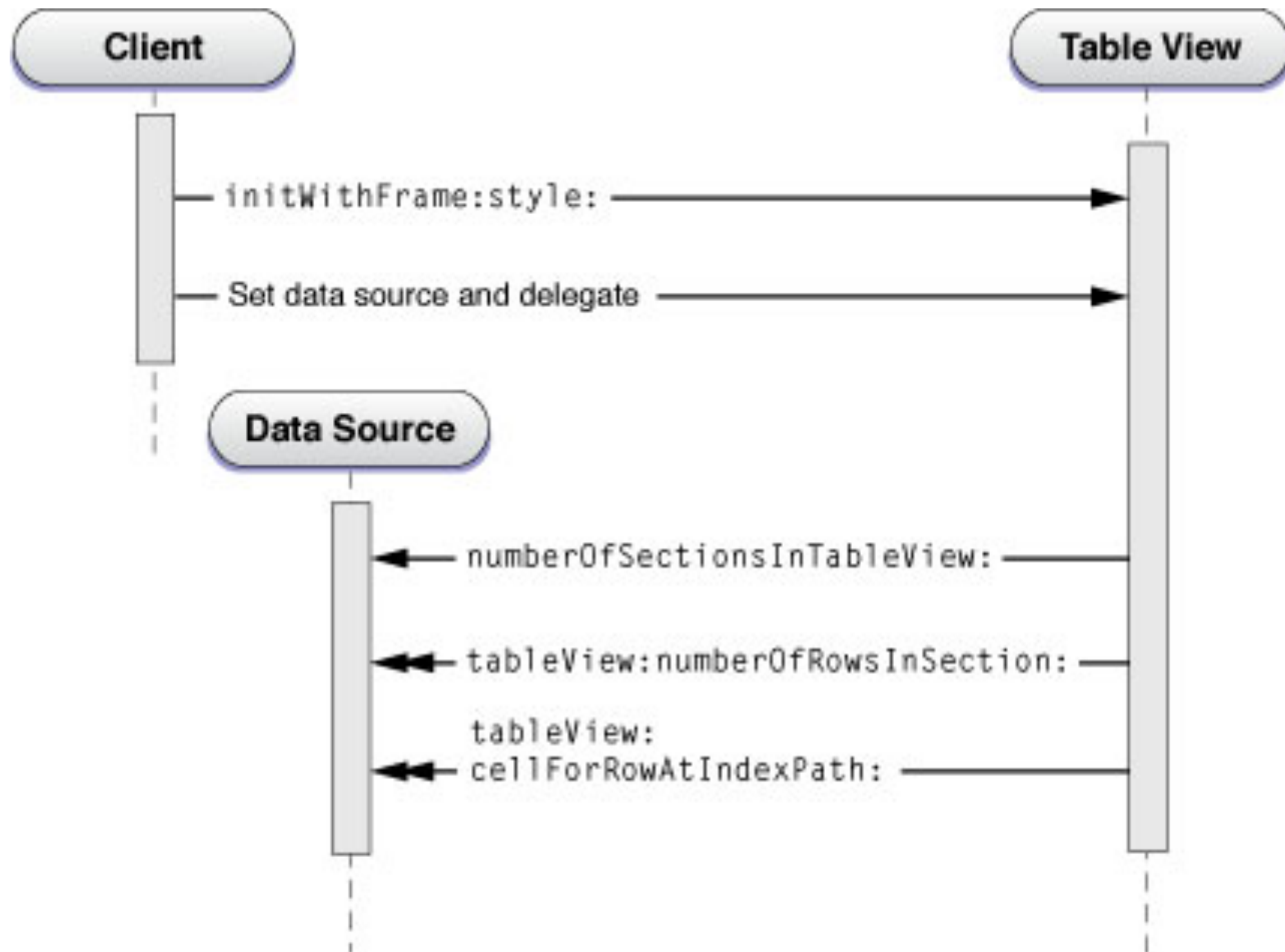
`UITableViewDataSource`

The UITableViewDataSource protocol is adopted by an object that mediates the application's data model for a `UITableView` object. The data source provides the table-view object with the information it needs to construct and modify a table view.

`UITableViewDataSourcePrefetching`

A protocol that provides advance warning of the data requirements for a table view, allowing the triggers of asynchronous data load operations.

Configuring a table view



UITableViewDataSource

```
protocol UITableViewDataSource : NSObjectProtocol {  
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int  
    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell  
}
```

```
func tableView(  
    _ tableView: UITableView,  
    numberOfRowsInSection section: Int  
) -> Int {  
    // 섹션별 row 갯수  
}
```

```
func tableView(  
    _ tableView: UITableView,  
    cellForRowAt indexPath: IndexPath  
) -> UITableViewCell {  
    // UITableViewCell 커스터마이징  
}
```

UITableViewDelegate

Interaction 과 관련된 동작을 커스터마이징하는 프로토콜

Customizing the Table View Behavior

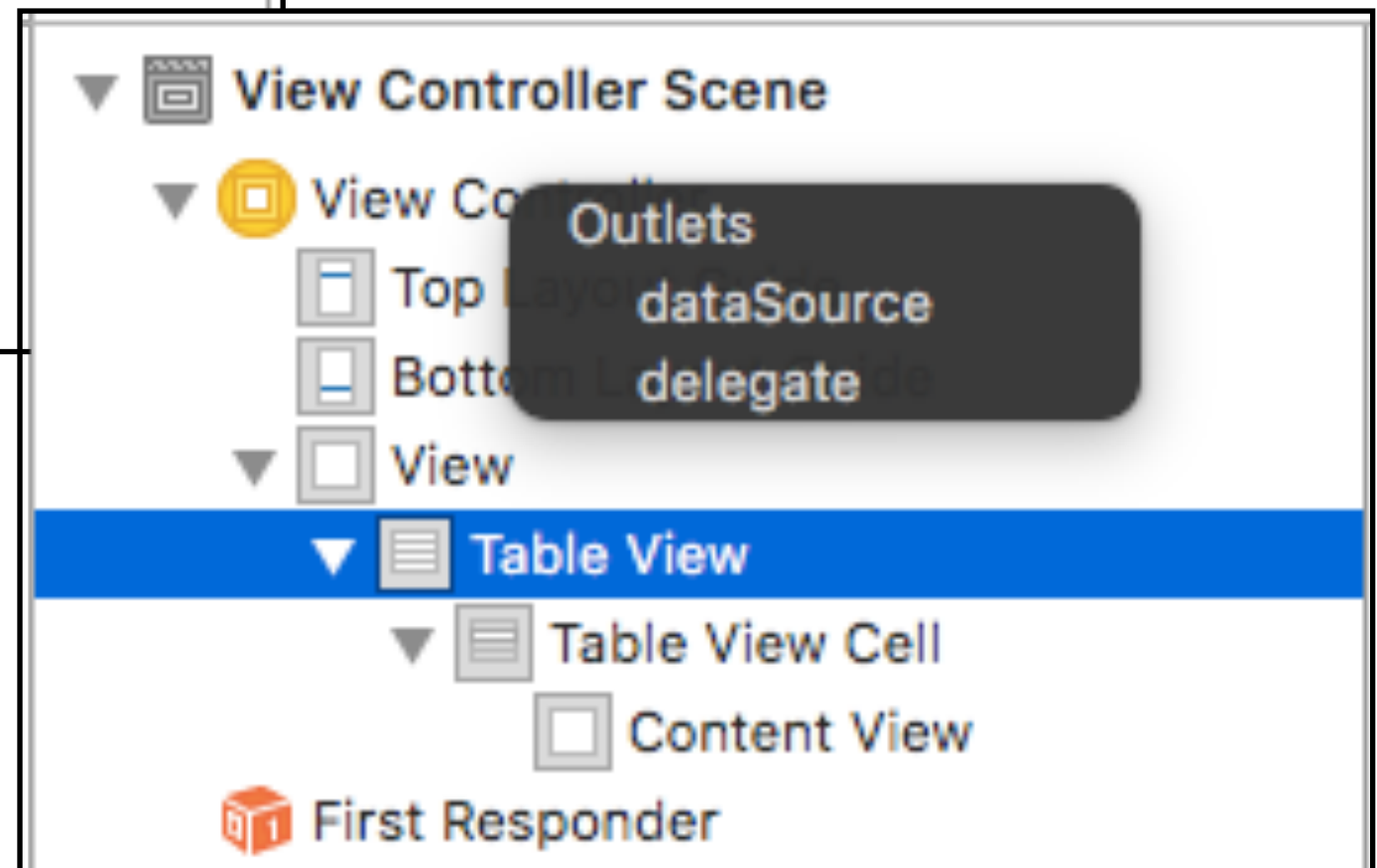
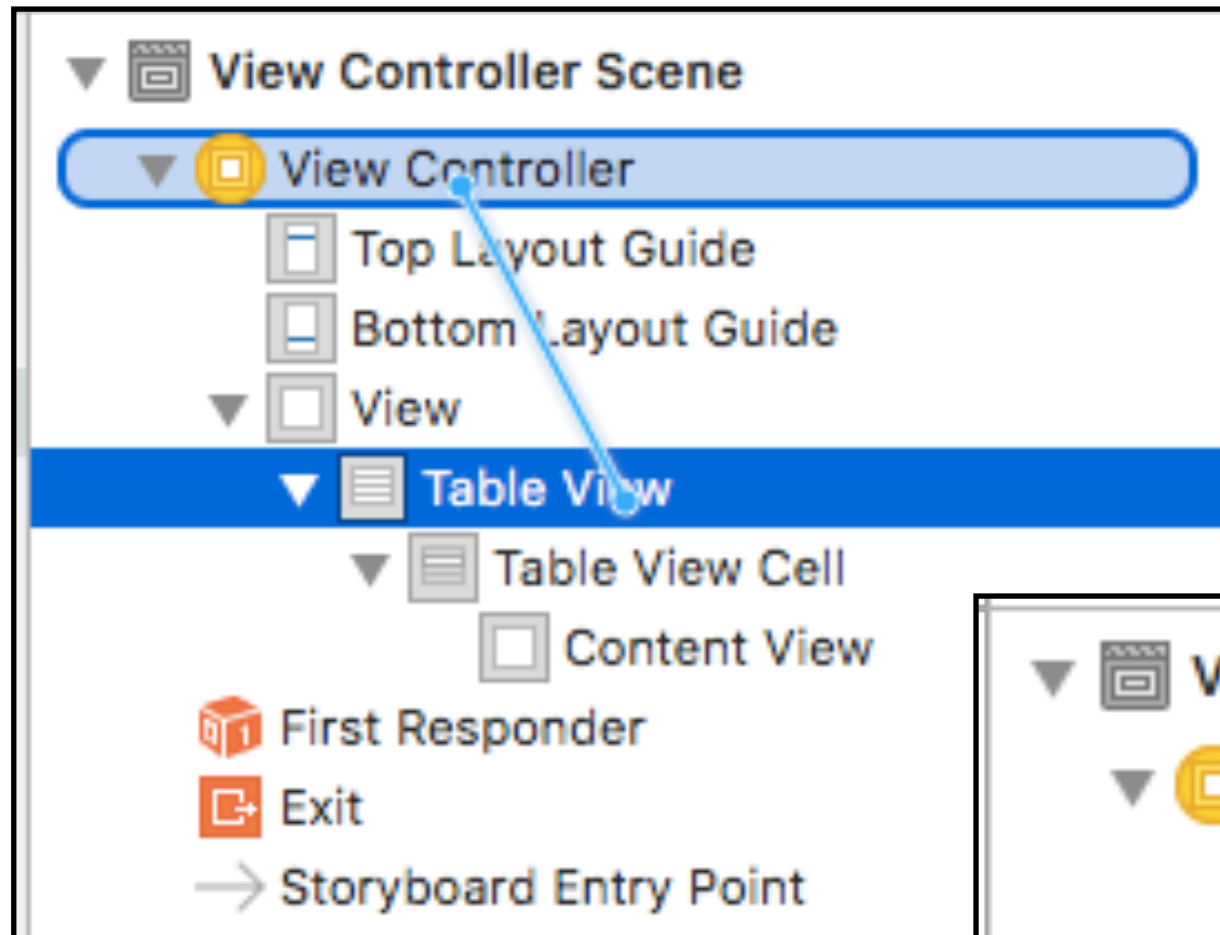
`delegate`

The object that acts as the delegate of the table view.

`UITableViewDelegate`

The delegate of a `UITableView` object must adopt the `UITableViewDelegate` protocol. Optional methods of the protocol allow the delegate to manage selections, configure section headings and footers, help to delete and reorder cells, and perform other actions.

DataSource & Delegate



UITableViewCell

TableView 에서 사용하는 각 아이템을 담는 컨테이너

반복되는 셀을 매번 생성, 해제, 재할당하는 부담을 덜기 위해 화면에 보이지 않는 부분의 셀을 재사용

Creating Table View Cells

- [registerNib:forCellReuseIdentifier:](#)

Registers a nib object containing a cell with the table view under a specified identifier.

- [registerClass:forCellReuseIdentifier:](#)

Registers a class for use in creating new table cells.

- [dequeueReusableCellWithIdentifier:forIndexPath:](#)

Returns a reusable table-view cell object for the specified reuse identifier and adds it to the table.

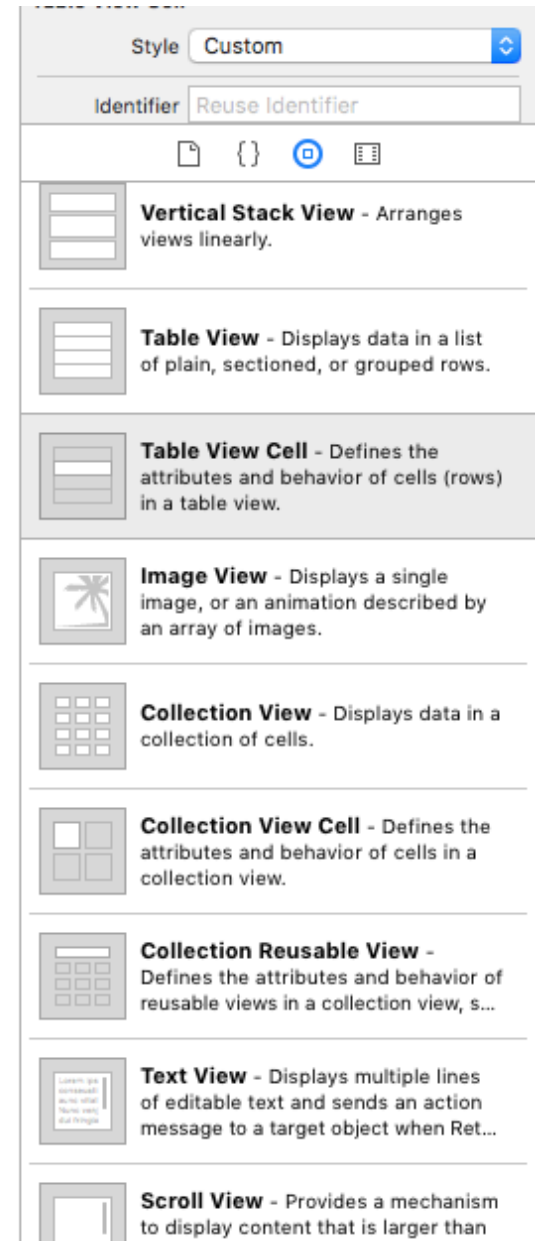
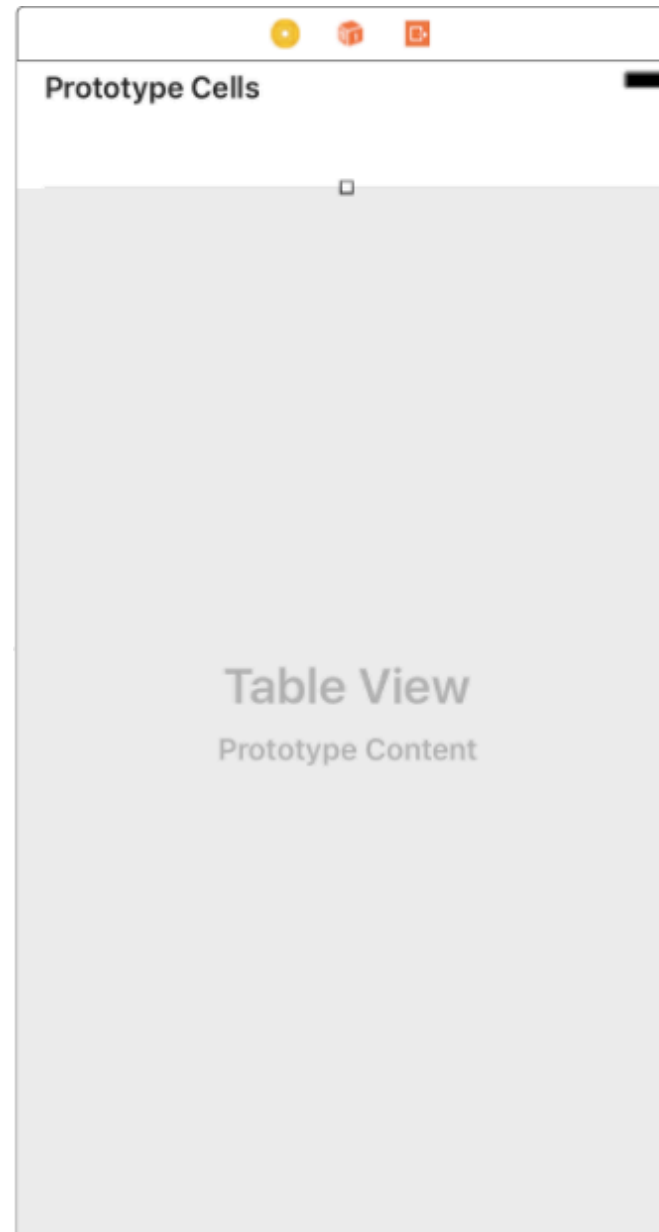
- [dequeueReusableCellWithIdentifier:](#)

Returns a reusable table-view cell object located by its identifier.

UITableViewCell

Storyboard

- TableView 를 뷰에 올린 후
오토레이아웃 적용
- UITableViewCell 을 TableView 에 추가



UITableViewCell Identifier

각 Cell 을 구분해서 재사용하기 위한 식별자

Table View Cell

Style Custom

Identifier Reuse Identifier

Selection Default

Accessory None

Editing Acc. None

Focus Style Default

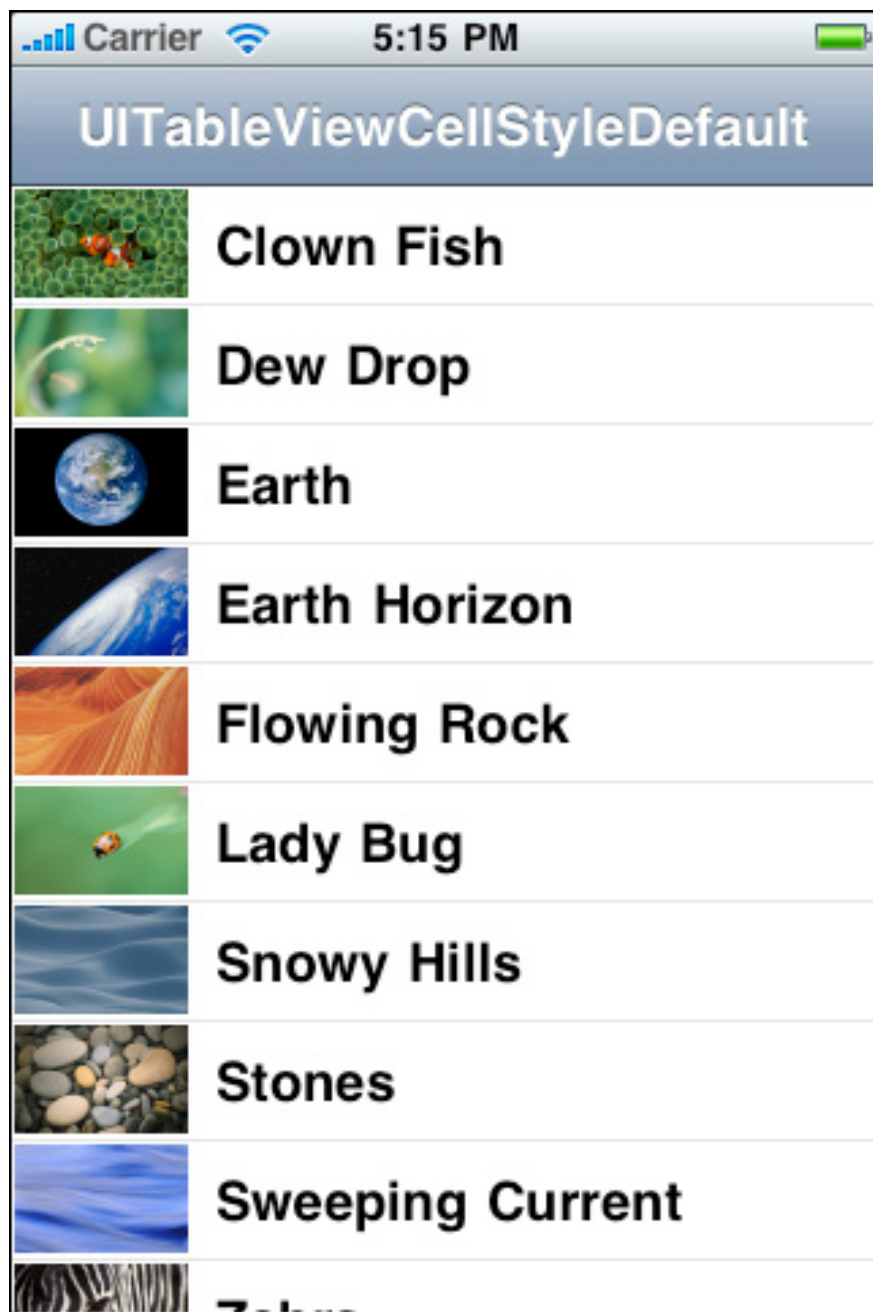
Indentation 0 10
Level Width

☒ Indent While Editing

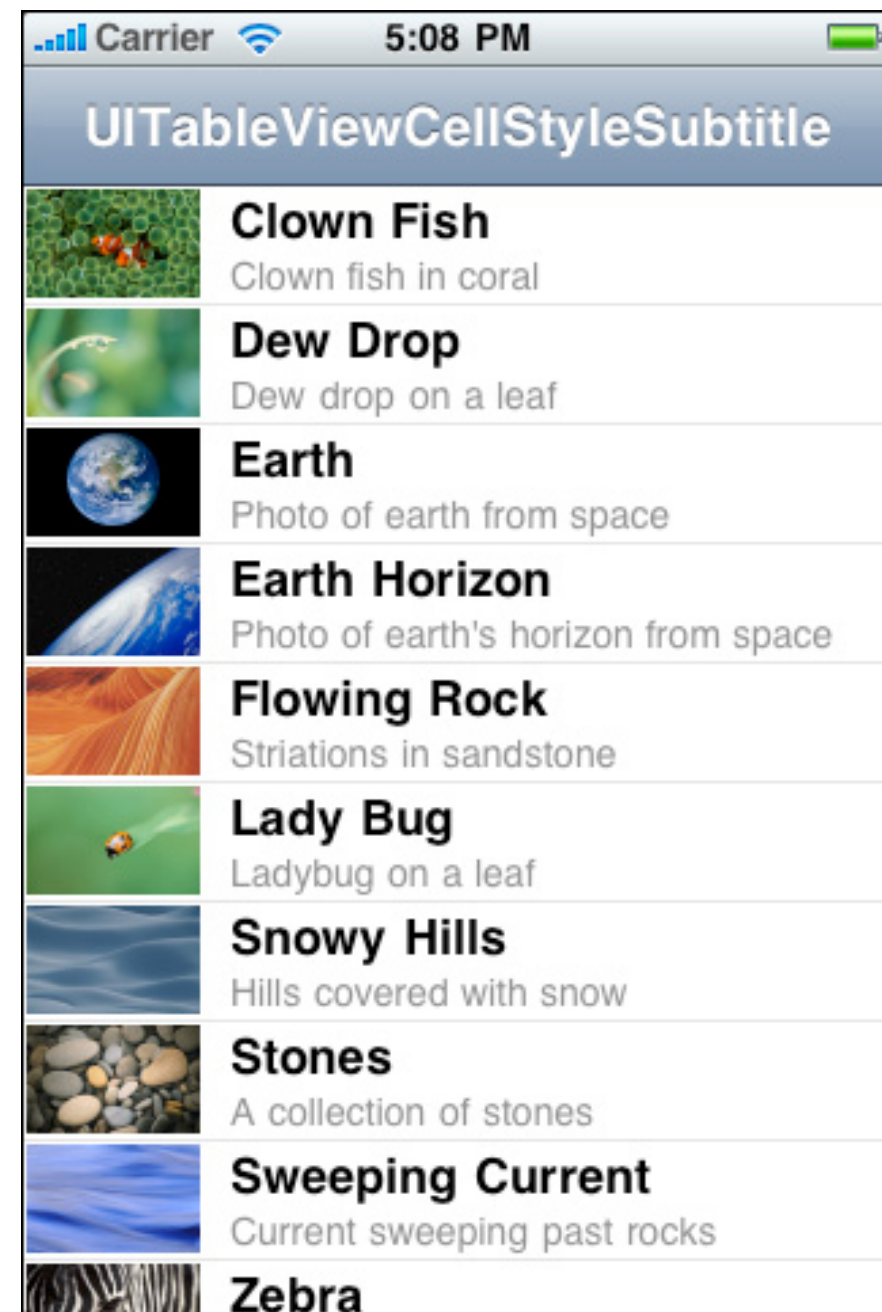
☐ Shows Re-order Controls

Separator Default Insets

UITableViewCell Style

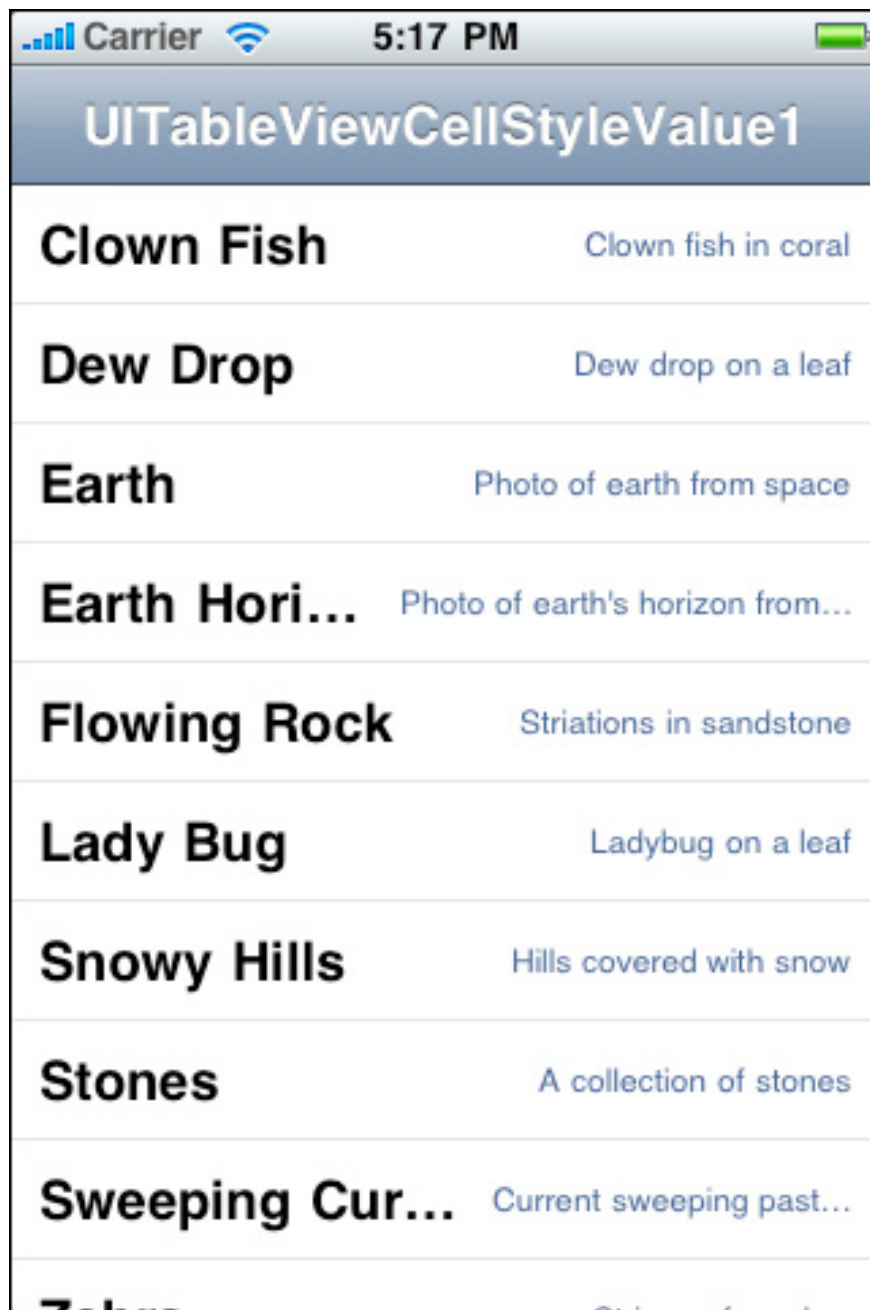


default

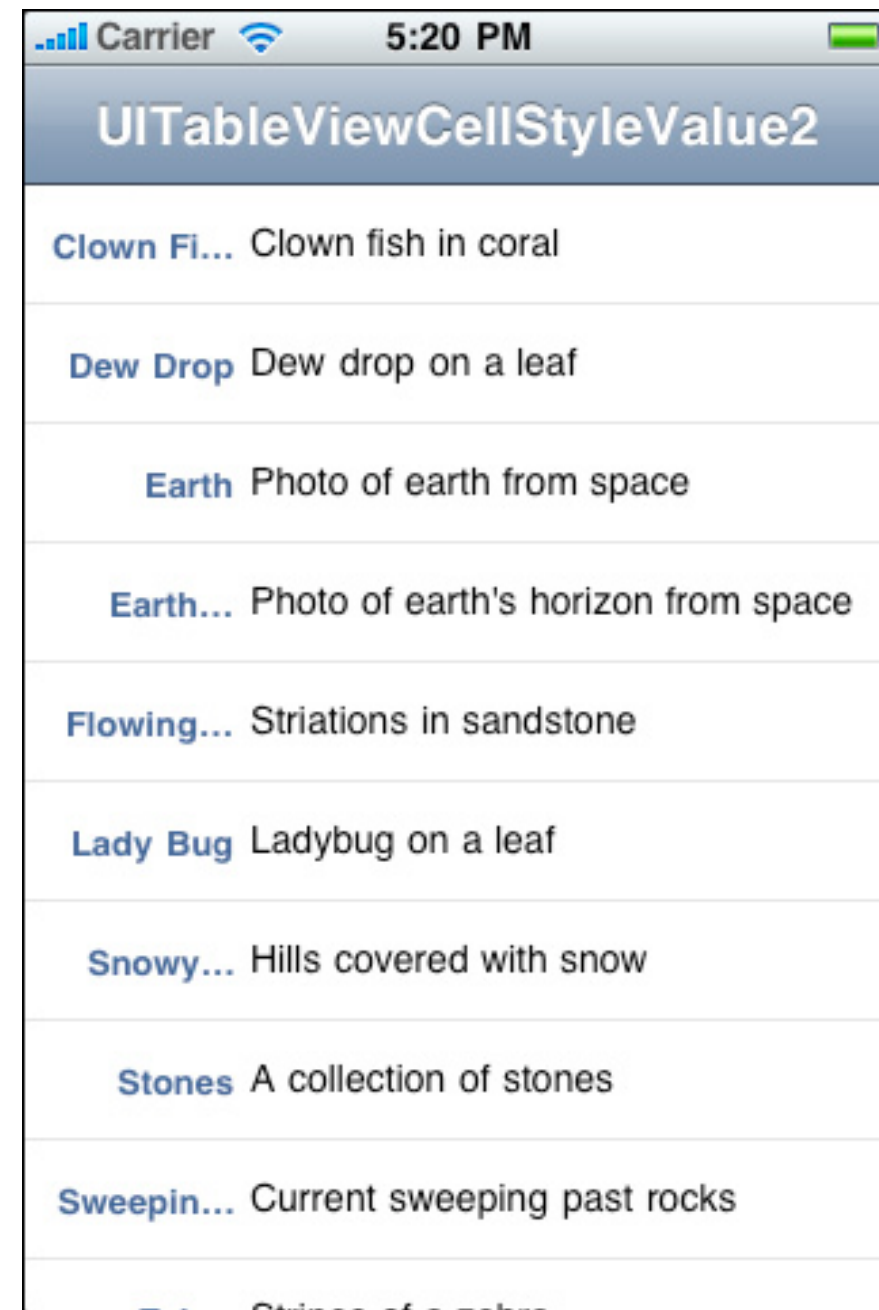


subtitle

UITableViewCell Style



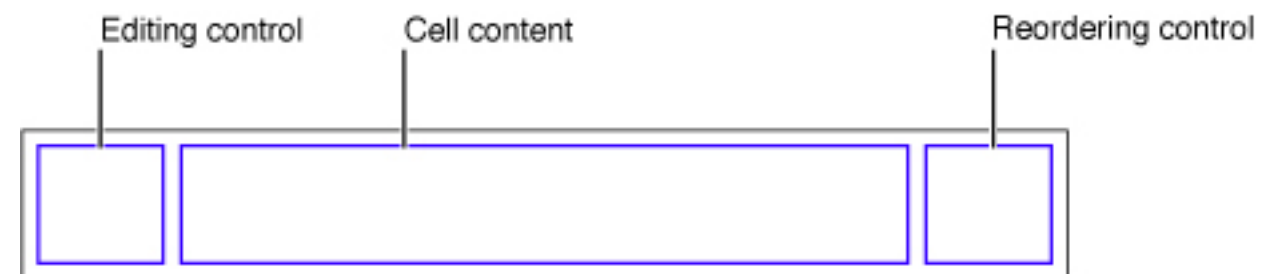
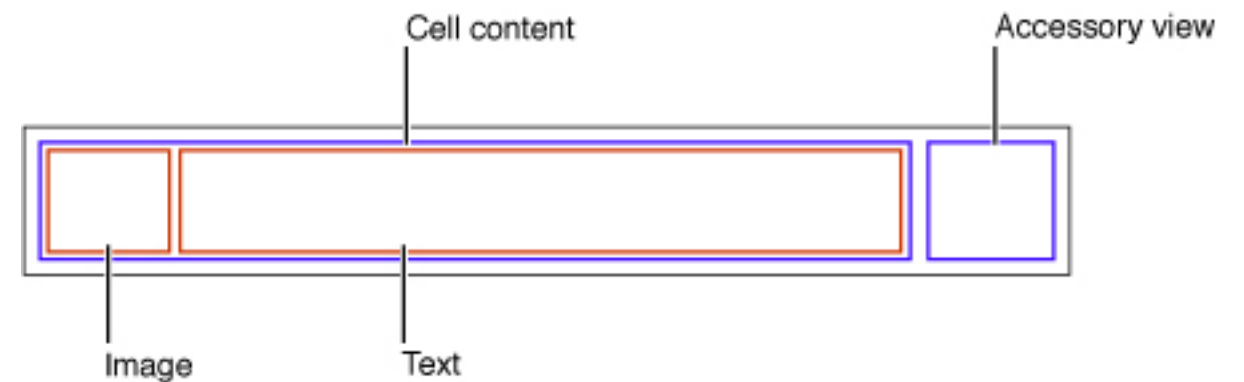
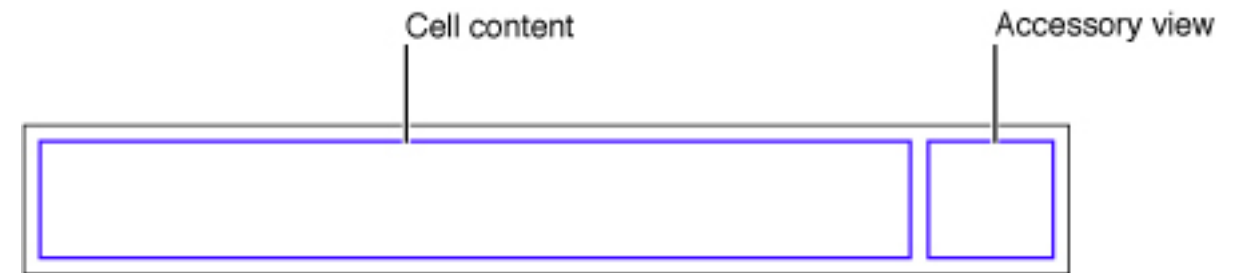
value 1






value 2

UITableViewCell

- contentView
- .textLabel
- detailTextLabel
- imageView
- accessoryView



UITableViewCell

	Bear	✓
	Black Swan	✓
	Buffalo	✓