

Class UITableView

Initializing a UITableView Object

```
init(frame: CGRect, style: UITableView.Style)
    Initializes and returns a table view object having the given frame and style.

init?(coder: NSCoder)
```

Providing the Table View Data

```
var dataSource: UITableViewDataSource?
    The object that acts as the data source of the table view.

protocol 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.

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

Customizing the Table View Behavior

```
var delegate: UITableViewDelegate?
    The object that acts as the delegate of the table view.

protocol 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.
```

Configuring a Table View

```
var style: UITableView.Style
    Returns the style of the table view.

func numberOfRows(inSection: Int) -> Int
    Returns the number of rows (table cells) in a specified section.

var numberOfSections: Int
    The number of sections in the table view.

var rowHeight: CGFloat
    The height of each row (that is, table cell) in the table view.

var separatorStyle: UITableViewCell.SeparatorStyle
    The style for table cells used as separators.

var separatorColor: UIColor?
    The color of separator rows in the table view.

var separatorEffect: UIVisualEffect?
    The effect applied to table separators.

var backgroundView: UIView?
    The background view of the table view.

var separatorInset: UIEdgeInsets
    Specifies the default inset of cell separators.

var separatorInsetReference: UITableView.SeparatorInsetReference
    An indicator of how the separator inset value should be interpreted.

enum UITableView.SeparatorInsetReference
    Constants indicating how to interpret the separator inset value of a table view.

var cellLayoutMarginsFollowReadableWidth: Bool
    A Boolean value that indicates whether the cell margins are derived from the width of the readable content guide.
```

Creating Table View Cells

```
func register(UINib?, forCellReuseIdentifier: String)
    Registers a nib object containing a cell with the table view under a specified identifier.

func register(AnyClass?, forCellReuseIdentifier: String)
    Registers a class for use in creating new table cells.

func dequeueReusableCell(withIdentifier: String, for: IndexPath) -
    > UITableViewCell
    Returns a reusable table-view cell object for the specified reuse identifier and adds it to the
    table.

func dequeueReusableCell(withIdentifier: String) -> UITableView
    Cell?
    Returns a reusable table-view cell object located by its identifier.
```

Accessing Header and Footer Views

```
func register(UINib?, forHeaderFooterViewReuseIdentifier: String)
    Registers a nib object containing a header or footer with the table view under a specified
    identifier.

func register(AnyClass?, forHeaderFooterViewReuseIdentifier:
String)
    Registers a class for use in creating new table header or footer views.

func dequeueReusableHeaderFooterView(withIdentifier: String) ->
UITableViewHeaderFooterView?
    Returns a reusable header or footer view located by its identifier.

var tableViewHeader: UIView?
    Returns an accessory view that is displayed above the table.

var tableViewFooter: UIView?
    Returns an accessory view that is displayed below the table.

var sectionHeaderHeight: CGFloat
    The height of section headers in the table view.

var sectionFooterHeight: CGFloat
    The height of section footers in the table view.

func headerView(forSection: Int) -> UITableViewHeaderFooterView?
    Returns the header view associated with the specified section.

func footerView(forSection: Int) -> UITableViewHeaderFooterView?
    Returns the footer view associated with the specified section.
```

Accessing Cells and Sections

```
func cellForRow(at: IndexPath) -> UITableViewCell?
    Returns the table cell at the specified index path.

func indexPath(for: UITableViewCell) -> IndexPath?
    Returns an index path representing the row and section of a given table-view cell.

func indexPathForRow(at: CGPoint) -> IndexPath?
    Returns an index path identifying the row and section at the given point.

func indexPathsForRows(in: CGRect) -> [IndexPath]?
    An array of index paths each representing a row enclosed by a given rectangle.

var visibleCells: [UITableViewCell]
    The table cells that are visible in the table view.

var indexPathsForVisibleRows: [IndexPath]?
    An array of index paths each identifying a visible row in the table view.
```

Estimating Element Heights

```
var estimatedRowHeight: CGFloat
    The estimated height of rows in the table view.

var estimatedSectionHeaderHeight: CGFloat
    The estimated height of section headers in the table view.

var estimatedSectionFooterHeight: CGFloat
    The estimated height of section footers in the table view.
```

Scrolling the Table View

```
func scrollToRow(at: IndexPath, at: UITableView.ScrollPosition,
    animated: Bool)
    Scrolls through the table view until a row identified by index path is at a particular location on the screen.

func scrollToNearestSelectedRow(at: UITableView.ScrollPosition,
    animated: Bool)
    Scrolls the table view so that the selected row nearest to a specified position in the table view is at that position.
```

Managing Selections

var `indexPathForSelectedRow`: IndexPath?

An index path identifying the row and section of the selected row.

var `indexPathsForSelectedRows`: [IndexPath]?

The index paths representing the selected rows.

func `selectRow(at: IndexPath?, animated: Bool, scrollPosition: UITableView.ScrollPosition)`

Selects a row in the table view identified by index path, optionally scrolling the row to a location in the table view.

func `deselectRow(at: IndexPath, animated: Bool)`

Deselects a given row identified by index path, with an option to animate the deselection.

var `allowsSelection`: Bool

A Boolean value that determines whether users can select a row.

var `allowsMultipleSelection`: Bool

A Boolean value that determines whether users can select more than one row outside of editing mode.

var `allowsSelectionDuringEditing`: Bool

A Boolean value that determines whether users can select cells while the table view is in editing mode.

var `allowsMultipleSelectionDuringEditing`: Bool

A Boolean value that controls whether users can select more than one cell simultaneously in editing mode.

Inserting, Deleting, and Moving Rows and Sections

func `insertRows(at: [IndexPath], with: UITableView.RowAnimation)`

Inserts rows in the table view at the locations identified by an array of index paths, with an option to animate the insertion.

func `deleteRows(at: [IndexPath], with: UITableView.RowAnimation)`

Deletes the rows specified by an array of index paths, with an option to animate the deletion.

func `moveRow(at: IndexPath, to: IndexPath)`

Moves the row at a specified location to a destination location.

func `insertSections(IndexSet, with: UITableView.RowAnimation)`

Inserts one or more sections in the table view, with an option to animate the insertion.

func `deleteSections(IndexSet, with: UITableView.RowAnimation)`

Deletes one or more sections in the table view, with an option to animate the deletion.

func `moveSection(Int, toSection: Int)`

Moves a section to a new location in the table view.

func `performBatchUpdates(() -> Void)?, completion: ((Bool) -> Void)? = nil)`

Animates multiple insert, delete, reload, and move operations as a group.

func `beginUpdates()`

Begins a series of method calls that insert, delete, or select rows and sections of the table view.

func `endUpdates()`

Concludes a series of method calls that insert, delete, select, or reload rows and sections of the table view.

Managing Drag Interactions

var `dragDelegate: UITableViewDragDelegate?`

The delegate object that manages the dragging of items from the table view.

protocol `UITableViewDragDelegate`

The interface for initiating drags from a table view.

var `hasActiveDrag: Bool`

A Boolean value indicating whether rows were lifted from the table view and have not yet been dropped.

var `dragInteractionEnabled: Bool`

A Boolean value indicating whether the table view supports drags and drops between apps.

Managing Drop Interactions

```
var dropDelegate: UITableViewDropDelegate?
    The delegate object that manages the dropping of content into the table view.
```

```
protocol UITableViewDropDelegate
    The interface for handling drops in a table view.
```

```
var hasActiveDrop: Bool
```

Managing the Editing of Table Cells

```
var isEditing: Bool
    A Boolean value that determines whether the table view is in editing mode.
```

```
func setEditing(Bool, animated: Bool)
    Toggles the table view into and out of editing mode.
```

Reloading the Table View

```
var hasUncommittedUpdates: Bool
    A Boolean value indicating whether the table view contains drop placeholders or is reordering its rows as part of handling a drop.
```

```
func reloadData()
    Reloads the rows and sections of the table view.
```

```
func reloadRows(at: [IndexPath], with: UITableView.RowAnimation)
    Reloads the specified rows using an animation effect.
```

```
func reloadSections(IndexSet, with: UITableView.RowAnimation)
    Reloads the specified sections using a given animation effect.
```

```
func reloadSectionIndexTitles()
    Reloads the items in the index bar along the right side of the table view.
```

Accessing Drawing Areas of the Table View

```
func rect(forSection: Int) -> CGRect
    Returns the drawing area for a specified section of the table view.
```

```
func rectForRow(at: IndexPath) -> CGRect
    Returns the drawing area for a row identified by index path.
```

```
func rectForFooter(inSection: Int) -> CGRect
    Returns the drawing area for the footer of the specified section.
```

```
func rectForHeader(inSection: Int) -> CGRect
    Returns the drawing area for the header of the specified section.
```

Prefetching Data

If your table view relies on an expensive data loading process, you can improve your user experience by prefetching data before it is needed for display. Assign an object that conforms to the [UITableViewDataSourcePrefetching](#) protocol to the [prefetchDataSource](#) property to receive notifications of when to prefetch data for cells.

var [prefetchDataSource](#): UITableViewDataSourcePrefetching?

The object that acts as the prefetching data source for the table view, receiving notifications of upcoming cell data requirements.

Configuring the Table Index

var [sectionIndexMinimumDisplayRowCount](#): Int

The number of table rows at which to display the index list on the right edge of the table.

var [sectionIndexColor](#): UIColor?

The color to use for the table view's index text.

var [sectionIndexBackgroundColor](#): UIColor?

The color to use for the background of the table view's section index while not being touched.

var [sectionIndexTrackingBackgroundColor](#): UIColor?

The color to use for the table view's index background area.

Managing Focus

var [remembersLastFocusedIndexPath](#): Bool

A Boolean value that indicates whether the table view should automatically return the focus to the cell at the last focused index path.

Constants

enum [UITableView.Style](#)

The style of the table view.

enum [UITableView.ScrollPosition](#)

The position in the table view (top, middle, bottom) to which a given row is scrolled.

enum [UITableView.RowAnimation](#)

The type of animation when rows are inserted or deleted.

 [Section Index Icons](#)

Requests icon to be shown in the section index of a table view.

 [Default Dimension](#)

The default value for a given dimension.

Notifications

class let `selectionDidChangeNotification`: NSNotification.Name
Posted when the selected row in the posting table view changes.

Class

UITableViewCell

Initializing a UITableViewCell Object

`init(style: UITableViewCell.CellStyle, reuseIdentifier: String?)`
Initializes a table cell with a style and a reuse identifier and returns it to the caller.

`init?(coder: NSCoder)`

Reusing Cells

var `reuseIdentifier`: String?
A string used to identify a cell that is reusable.

func `prepareForReuse()`
Prepares a reusable cell for reuse by the table view's delegate.

Managing the Predefined Content

var `textLabel`: UILabel?
Returns the label used for the main textual content of the table cell.

var `detailTextLabel`: UILabel?
Returns the secondary label of the table cell if one exists.

var `imageView`: UIImageView?
Returns the image view of the table cell.

Accessing Views of the Cell Object

- var `contentView`: UIView
Returns the content view of the cell object.
- var `backgroundView`: UIView?
The view used as the background of the cell.
- var `selectedBackgroundView`: UIView?
The view used as the background of the cell when it is selected.
- var `multipleSelectionBackgroundView`: UIView?
The background view to use for a selected cell when the table view allows multiple row selections.

Managing Accessory Views

- var `accessoryType`: UITableViewCell.AccessoryType
The type of standard accessory view the cell should use (normal state).
- var `accessoryView`: UIView?
A view that is used, typically as a control, on the right side of the cell (normal state).
- var `editingAccessoryType`: UITableViewCell.AccessoryType
The type of standard accessory view the cell should use in the table view's editing state.
- var `editingAccessoryView`: UIView?
A view that is used typically as a control on the right side of the cell when it is in editing mode.

Managing Cell Selection and Highlighting

- var `isSelected`: Bool
A Boolean value that indicates whether the cell is selected.
- var `selectionStyle`: UITableViewCell.SelectionStyle
The style of selection for a cell.
- func `setSelected`(Bool, `animated`: Bool)
Sets the selected state of the cell, optionally animating the transition between states.
- var `isHighlighted`: Bool
A Boolean value that indicates whether the cell is highlighted.
- func `setHighlighted`(Bool, `animated`: Bool)
Sets the highlighted state of the cell, optionally animating the transition between states.

Editing the Cell

```
var isEditing: Bool
    A Boolean value that indicates whether the cell is in an editable state.

func setEditing(Bool, animated: Bool)
    Toggles the receiver into and out of editing mode.

var editingStyle: UITableViewCell.EditingStyle
    The editing style of the cell.

var showingDeleteConfirmation: Bool
    Returns whether the cell is currently showing the delete-confirmation button.

var showsReorderControl: Bool
    A Boolean value that determines whether the cell shows the reordering control.
```

Dragging the Row

```
var userInteractionEnabledWhileDragging: Bool
    A Boolean value indicating whether users can interact with a cell while it is being dragged.

func dragStateDidChange(UITableViewCell.DragState)
    Notifies the cell that its drag status changed.

enum UITableViewCell.DragState
    Constants indicating the current state of a row involved in a drag operation.
```

Adjusting to State Transitions

```
func willTransition(to: UITableViewCell.StateMask)
    Called on the cell just before it transitions between cell states.

func didTransition(to: UITableViewCell.StateMask)
    Called on the cell just after it transitions between cell states.
```

Managing Content Indentation

var `indentationLevel`: Int
The indentation level of the cell's content.

var `indentationWidth`: CGFloat
The width for each level of indentation of a cell's content.

var `shouldIndentWhileEditing`: Bool
A Boolean value that controls whether the cell background is indented when the table view is in editing mode.

var `separatorInset`: UIEdgeInsets
The inset values for the cell's content.

Managing Focus

var `focusStyle`: UITableViewCell.FocusStyle
The appearance of the cell when focused.

Constants

enum `UITableViewCell.CellStyle`
An enumeration for the various styles of cells.

enum `UITableViewCell.SelectionStyle`
The style of selected cells.

enum `UITableViewCell.EditingStyle`
The editing control used by a cell.

enum `UITableViewCell.AccessoryType`
The type of standard accessory control used by a cell.

struct `UITableViewCell.StateMask`
Constants used to determine the new state of a cell as it transitions between states.

enum `UITableViewCell.SeparatorStyle`
The style for cells used as separators.

enum `UITableViewCell.FocusStyle`
The style of focused cells.

Protocol UITableViewDelegate

Configuring Rows for the Table View

func tableView(UITableView, heightForRowAt: IndexPath) -> CGFloat
Asks the delegate for the height to use for a row in a specified location.

func tableView(UITableView, estimatedHeightForRowAt: IndexPath) -> CGFloat
Asks the delegate for the estimated height of a row in a specified location.

func tableView(UITableView, indentationLevelForRowAt: IndexPath) -> Int
Asks the delegate to return the level of indentation for a row in a given section.

func tableView(UITableView, willDisplay: UITableViewCell, forRowAt: IndexPath)
Tells the delegate the table view is about to draw a cell for a particular row.

func tableView(UITableView, shouldSpringLoadRowAt: IndexPath, with: UISpringLoadedInteractionContext) -> Bool
Called to let you fine tune the spring-loading behavior of the rows in a table.

Managing Accessory Views

func tableView(UITableView, editActionsForRowAt: IndexPath) -> [UITableViewRowAction]?
Asks the delegate for the actions to display in response to a swipe in the specified row.

func tableView(UITableView, accessoryButtonTappedForRowWith: IndexPath)
Tells the delegate that the user tapped the accessory (disclosure) view associated with a given row.

Managing Selections

func tableView(UITableView, willSelectRowAt: IndexPath) -> IndexPath?
Tells the delegate that a specified row is about to be selected.

func tableView(UITableView, didSelectRowAt: IndexPath)
Tells the delegate that the specified row is now selected.

func tableView(UITableView, willDeselectRowAt: IndexPath) -> IndexPath?
Tells the delegate that a specified row is about to be deselected.

func tableView(UITableView, didDeselectRowAt: IndexPath)
Tells the delegate that the specified row is now deselected.

Modifying the Header and Footer of Sections

func tableView(UITableView, viewForHeaderInSection: Int) -> UIView?
Asks the delegate for a view object to display in the header of the specified section of the table view.

func tableView(UITableView, viewForFooterInSection: Int) -> UIView?
Asks the delegate for a view object to display in the footer of the specified section of the table view.

func tableView(UITableView, heightForHeaderInSection: Int) -> CGFloat
Asks the delegate for the height to use for the header of a particular section.

func tableView(UITableView, estimatedHeightForHeaderInSection: Int) -> CGFloat
Asks the delegate for the estimated height of the header of a particular section.

func tableView(UITableView, heightForFooterInSection: Int) -> CGFloat
Asks the delegate for the height to use for the footer of a particular section.

func tableView(UITableView, estimatedHeightForFooterInSection: Int) -> CGFloat
Asks the delegate for the estimated height of the footer of a particular section.

func tableView(UITableView, willDisplayHeaderView: UIView, forSection: Int)
Tells the delegate that a header view is about to be displayed for the specified section.

func tableView(UITableView, willDisplayFooterView: UIView, forSection: Int)
Tells the delegate that a footer view is about to be displayed for the specified section.

Editing Table Rows

func tableView(UITableView, willBeginEditingRowAt: IndexPath)
Tells the delegate that the table view is about to go into editing mode.

func tableView(UITableView, didEndEditingRowAt: IndexPath?)
Tells the delegate that the table view has left editing mode.

func tableView(UITableView, editingStyleForRowAt: IndexPath) -> UITableViewCell.EditingStyle
Asks the delegate for the editing style of a row at a particular location in a table view.

func tableView(UITableView, titleForDeleteConfirmationButtonForRowAt: IndexPath) -> String?
Changes the default title of the delete-confirmation button.

func tableView(UITableView, shouldIndentWhileEditingRowAt: IndexPath) -> Bool
Asks the delegate whether the background of the specified row should be indented while the table view is in editing mode.

Reordering Table Rows

func tableView(UITableView, targetIndexPathForMoveFromRowAt: IndexPath, toProposedIndexPath: IndexPath) -> IndexPath
Asks the delegate to return a new index path to retarget a proposed move of a row.

Tracking the Removal of Views

func tableView(UITableView, didEndDisplaying: UITableViewCell, forRowAt: IndexPath)
Tells the delegate that the specified cell was removed from the table.

func tableView(UITableView, didEndDisplayingHeaderView: UIView, forSection: Int)
Tells the delegate that the specified header view was removed from the table.

func tableView(UITableView, didEndDisplayingFooterView: UIView, forSection: Int)
Tells the delegate that the specified footer view was removed from the table.

Copying and Pasting Row Content

```
func tableView(UITableView, shouldShowMenuForRowAt: IndexPath) -> Bool
```

Asks the delegate if the editing menu should be shown for a certain row.

```
func tableView(UITableView, canPerformAction: Selector, forRowAt: IndexPath, withSender: Any?) -> Bool
```

Asks the delegate if the editing menu should omit the Copy or Paste command for a given row.

```
func tableView(UITableView, performAction: Selector, forRowAt: IndexPath, withSender: Any?)
```

Tells the delegate to perform a copy or paste operation on the content of a given row.

Managing Table View Highlighting

```
func tableView(UITableView, shouldHighlightRowAt: IndexPath) -> Bool
```

Asks the delegate if the specified row should be highlighted.

```
func tableView(UITableView, didHighlightRowAt: IndexPath)
```

Tells the delegate that the specified row was highlighted.

```
func tableView(UITableView, didUnhighlightRowAt: IndexPath)
```

Tells the delegate that the highlight was removed from the row at the specified index path.

Managing Table View Focus

```
func tableView(UITableView, canFocusRowAt: IndexPath) -> Bool
```

Asks the delegate whether the cell at the specified index path is itself focusable.

```
func tableView(UITableView, shouldUpdateFocusIn: UITableViewFocus UpdateContext) -> Bool
```

Asks the delegate whether the focus update specified by the context is allowed to occur.

```
func tableView(UITableView, didUpdateFocusIn: UITableViewFocus UpdateContext, with: UIFocusAnimationCoordinator)
```

Tells the delegate that a focus update specified by the context has just occurred.

```
func indexPathForPreferredFocusedView(in: UITableView) -> IndexPath?
```

Asks the delegate for the table view's index path for the preferred focused view.

Handling Swipe Actions

```
func tableView(UITableView, leadingSwipeActionsConfigurationForRow
At: IndexPath) -> UISwipeActionsConfiguration?
```

Returns the swipe actions to display on the leading edge of the row.

```
func tableView(UITableView, trailingSwipeActionsConfigurationForRow
At: IndexPath) -> UISwipeActionsConfiguration?
```

Returns the swipe actions to display on the trailing edge of the row.]

Protocol

UITableViewDataSource

Configuring a Table View

```
func tableView(UITableView, cellForRowAt: IndexPath) -> UITableView
Cell
```

Asks the data source for a cell to insert in a particular location of the table view.

Required.

```
func numberOfSections(in: UITableView) -> Int
```

Asks the data source to return the number of sections in the table view.

```
func tableView(UITableView, numberOfRowsInSection: Int) -> Int
```

Tells the data source to return the number of rows in a given section of a table view.

Required.

```
func tableView(UITableView, titleForHeaderInSection: Int) ->
String?
```

Asks the data source for the title of the header of the specified section of the table view.

```
func tableView(UITableView, titleForFooterInSection: Int) ->
String?
```

Asks the data source for the title of the footer of the specified section of the table view.

Inserting or Deleting Table Rows

```
func tableView(UITableView, commit: UITableViewCellEditingStyle,
forRowAt: IndexPath)
```

Asks the data source to commit the insertion or deletion of a specified row in the receiver.

```
func tableView(UITableView, canEditRowAt: IndexPath) -> Bool
```

Asks the data source to verify that the given row is editable.

Reordering Table Rows

```
func tableView(UITableView, canMoveRowAt: IndexPath) -> Bool
```

Asks the data source whether a given row can be moved to another location in the table view.

```
func tableView(UITableView, moveRowAt: IndexPath, to: IndexPath)
```

Tells the data source to move a row at a specific location in the table view to another location.

Configuring an Index

```
func sectionIndexTitles(for: UITableView) -> [String]?
```

Asks the data source to return the titles for the sections for a table view.

```
func tableView(UITableView, sectionForSectionIndexTitle: String, at: Int) -> Int
```

Asks the data source to return the index of the section having the given title and section title index.

Protocol

UITableViewDataSourcePrefetching

Managing Data Prefetching

```
func tableView(UITableView, prefetchRowsAt: [IndexPath])
```

Instructs your prefetch data source object to begin preparing data for the cells at the supplied index paths.

Required.

```
func tableView(UITableView, cancelPrefetchingForRowsAt: [IndexPath])
```

Cancels a previously triggered data prefetch request.

Protocol

UITableView

DragDelegate

Providing the Items to Drag

```
func tableView(UITableView, itemsForBeginning: UIDragSession, at: IndexPath) -> [UIDragItem]
```

Provides the initial set of items (if any) to drag.

Required.

```
func tableView(UITableView, itemsForAddingTo: UIDragSession, at: IndexPath, point: CGPoint) -> [UIDragItem]
```

Adds the specified items to an existing drag session.

Tracking the Drag Session

```
func tableView(UITableView, dragSessionWillBegin: UIDragSession)
```

Signals the start of a drag operation involving content from the specified table view.

```
func tableView(UITableView, dragSessionDidEnd: UIDragSession)
```

Signals the end of a drag operation involving content from the specified table view

Providing a Custom Preview

```
func tableView(UITableView, dragPreviewParametersForRowAt: IndexPath) -> UIDragPreviewParameters?
```

Returns custom information about how to display the row at the specified location during the drag.

Instance Methods

```
func tableView(UITableView, dragSessionAllowsMoveOperation: UIDragSession) -> Bool
```

```
func tableView(UITableView, dragSessionIsRestrictedToDraggingApplication: UIDragSession) -> Bool
```

Protocol

UITableViewDropDelegate

Declaring Support for Handling Drops

```
func tableView(UITableView, canHandle: UIDropSession) -> Bool
```

Asks your delegate whether it can accept the specified type of data.

Incorporating the Dropped Data

```
func tableView(UITableView, performDropWith: UITableViewDropCoordinator)
```

Incorporates the dropped data into your data structures and updates the table.
Required.

Tracking the Drag Movements

```
func tableView(UITableView, dropSessionDidUpdate: UIDropSession, withDestinationIndexPath: IndexPath?) -> UITableViewDropProposal
```

Proposes how to handle a drop at the specified location in the table view.

```
func tableView(UITableView, dropSessionDidEnter: UIDropSession)
```

Called when dragged content enters the table view's bounds rectangle.

```
func tableView(UITableView, dropSessionDidExit: UIDropSession)
```

Called when dragged content exits the table view's bounds rectangle.

```
func tableView(UITableView, dropSessionDidEnd: UIDropSession)
```

Called to notify you when the drag operation ends.

Instance Methods

```
func tableView(UITableView, dropPreviewParametersForRowAt: IndexPath) -> UIDragPreviewParameters?
```

Protocol

UITableViewDropCoordinator

Getting the Dragged Items

```
var items: [UITableViewDropItem]
```

The items being dragged.

Required.

Getting the Drop Location

```
var destinationIndexPath: IndexPath?
```

The index path at which to insert the item into the table view.

Required.

Animating Rows to Their Destination

func **drop**(UIDragItem, **toRowAt**: IndexPath) -> UIDragAnimating
 Animates the item to the specified index path in the table view.
Required.

func **drop**(UIDragItem, **intoRowAt**: IndexPath, **rect**: CGRect) -> UIDragAnimating
Required.

func **drop**(UIDragItem, **to**: UIDragPreviewTarget) -> UIDragAnimating
 Animates the item to an arbitrary location in your view hierarchy.
Required.

Getting the Session Information

var **session**: UIDropSession
 The drop session containing information about the transaction.
Required.

var **proposal**: UITableViewDropProposal
 The proposal for how to incorporate the dropped items.
Required.

Instance Methods

func **drop**(UIDragItem, **to**: UITableViewDropPlaceholder) -> UITableViewDropPlaceholderContext
Required.

Class

UITableViewDropPlaceholder

Instance Properties

var **previewParametersProvider**: ((UITableViewCell) -> UIDragPreviewParameters)?

Class

UITableViewDropProposal

Creating a Drop Proposal

init(**operation**: UIDropOperation, **intent**: UITableViewDropProposal.Intent)
 Creates a drop proposal object that specifies how to incorporate the dropped content.

Getting the Proposed Drop Location

var `intent`: UITableViewDropProposal.Intent

The option to use when incorporating dropped items into your content.

enum `UITableViewDropProposal.Intent`

Constants indicating how you intend to handle a drop.

enum `UIDropOperation`

Operation types that determine how a drag and drop activity resolves when the user drops a drag item.

Protocol

UITableViewDropItem

Getting the Drag Item

var `dragItem`: UIDragItem

The item that was dragged.

Required.

Getting the Item Information

var `previewSize`: CGSize

The size of the drag item's preview.

Required.

var `sourceIndexPath`: IndexPath?

The index path of the item in the table view, if any.

Required.

Protocol

UITableViewDropPlaceholderContext

Updating the Placeholder Cell

func `commitInsertion(dataSourceUpdates: (IndexPath) -> Void) -> Bool`

Exchanges the placeholder cell for a cell with the final content.

Required.

Removing the Placeholder Cell

```
func deletePlaceholder() -> Bool
```

Removes an unneeded placeholder cell altogether from the table view.

Required.

Getting the Drag Item

```
var dragItem: UIDragItem
```

The drag item being represented by the placeholder cell.

Required.

Protocol

UIDataSourceTranslating

Instance Methods

```
func dataSourceIndexPath(forPresentationIndexPath: IndexPath?) -> IndexPath?
```

Required.

```
func dataSourceSectionIndex(forPresentationSectionIndex: Int) -> Int
```

Required.

```
func performUsingPresentationValues(() -> Void)
```

Required.

```
func presentationIndexPath(forDataSourceIndexPath: IndexPath?) -> IndexPath?
```

Required.

```
func presentationSectionIndex(forDataSourceSectionIndex: Int) -> Int
```

Required.

Class

UITableViewPlaceholder

Initializers

```
init(insertionIndexPath: IndexPath, reuseIdentifier: String, rowHeight: CGFloat)
```

Instance Properties

```
var cellUpdateHandler: ((UITableViewCell) -> Void)?
```


