Motivation

Basics

- manage common table tasks
- support section consisting of individual objects in each row
- support section consisting of rows populated from an array
- support edit, move, delete, add of rows in sections
- support moving rows between sections
- support disclosure accessory to automatically create and push controller
- observe changes to row's object and update cell automatically
- observe changes to array and update section automatically
- user's changes to row applied to object it represents
- user's changes to rows in section applied to array it represents

Sections Have

- header
- footer
- title
- content (array of objects representing rows)
- bindings (internal array, one object per content object)
- default moveable, deletable, addable flags for rows
- default cell class for rows
- default disclosure class for rows

Rows Have

- object they are bound to
- binding object made internally
- cell object (created on demand) to view/edit the object
- cell class used to determine kind of cell to make
- disclosure class used to determine kind of controller to make when disclosing
- label - used by cell to show item
- value - optionally used by cell to show and/or edit value
- moveable, deletable properties

Design

Section

- set up a section by either
- - add rows one at a time
- - set the content array

Adding Invidual Rows

- supply object
- supply properties describing the binding
- content array created automatically if necessary

Adding Arrays

- supply array containing raw values (strings, custom classes, whatever)
- supply properties describing the binding to be made for each item
- section registers as observer of the array

Binding Objects

- each one represents row
- binds an external object to the cell for the row
- can supply a label and a value to the cell
- label can be set manually, or use object value
- value can be the mapped object, or value of a keypath on it
- can specify cell class used to view/edit cell
- can specify view controller class used to disclose the item
- cell editing edits the object's value, not the object itself
- can mark items as moveable, deletable

Cells

- essentially any cell can be used
- must implement custom protocol, so cells must be subclassed
- standard classes supplied