AdvTableView

Overview: AdvTableView is a SwiftUI view that provides advanced table functionality with features like sorting, filtering, and toolbar actions. It is designed to work with data conforming to the IDataSet protocol and a view model conforming to the IViewModel protocol.

Usage: To use AdvTableView, initialize it with the necessary parameters, including bindings for items, selected items, and other configuration options. You can customize the table columns and appearance by providing a content closure containing AdvTableColumn elements.

Example Code

```
AdvTableView(items: $items, selectedItems: $selectedItems, viewModel: vm) {
    AdvTableColumn(header: "Column1", field: \Item.property1, size:
.flexible(minimum: 100, maximum: 200))
    AdvTableColumn(header: "Column2", field: \Item.property2, size: .fixed(150))
    // Add more columns as needed
}
```

Parameters:

- items: Binding to the array of data (rows) to be displayed in the table.
- selectedItems: Binding to the set of selected table rows.
- searchText: Binding to a string specifying the text to be searched in the selected column.
- isInEditState: Binding to a boolean determining whether the table is in edit mode.
- showHeader: Boolean determining whether the table header should be displayed (default: true).
- headerHeight: Height of the table header (default: 35).
- checkable: Boolean determining whether table rows can be selected via checkbox (default: false).
- editView: ViewContext specifying which view should be used to edit a table row (default: nil).
- viewModel: The ViewModel connected to the table (default: AdvViewModel<User>).
- filter: A closure for custom filtering of items (default: nil).
- borderWidth: Width of the table border (default: 0).
- rowHeight: Height of each table row (default: 30).
- hideToolbar: Boolean specifying whether the toolbar should be displayed (default: false).
- showButton: ShowDefaultButton determining which toolbar buttons are displayed (default: [.modifyViews, .update, .lookup]).
- additionalToolbarItems: An array of NavigationItems to extend the toolbar (default: []).
- showColoredBackground: Boolean specifying whether the table background is displayed in alternating colors (default: true).

- showColumns: Boolean determining whether the background displays the columns (default: true).
- lookupViewContext: ViewContext determining which ViewContext is used for a LookupView (default: nil).
- content: Closure containing the table content in the form of AdvTableColumns.

Example Usage

```
struct ContentView: View {
   // Replace YourItemType with the actual type of your data

    @State private var items: [YourItemType] = []
    @State private var selectedItems: Set<YourItemType> = []

    var body: some View {
        AdvTableView(items: $items, selectedItems: $selectedItems) {
            AdvTableColumn(header: "Column1", field: \YourItemType.property1, size:
             .flexible(minimum: 100, maximum: 200))
            AdvTableColumn(header: "Column2", field: \YourItemType.property2, size:
             .fixed(150))
            // Add more columns as needed
        }
    }
}
```

This is a basic example, and you may need to replace YourItemType with the actual type of your data. Additionally, you should customize the columns according to your data model.
Note: The code provided assumes the existence of certain protocols (IDataSet, IViewModel, etc.) and types (AdvTableColumn, AdvViewModel, etc.) that are not included in the provided code snippet. Please make sure to include the necessary implementations for these types in your project.

AdvTableColumn

Overview

The AdvTableColumn struct represents an advanced table column and is available starting from macOS 14.0 and iOS 17.0. It is designed to be used in conjunction with advanced table views for displaying data in SwiftUI applications.

Example Code

```
public init(header: String, field: WritableKeyPath<T, D>,
            size: GridItem.Size = .flexible(minimum: 10, maximum: 200),
            hasFocus: Bool = false, subType: SubDataType = .none,
            pickerColumn: (any IAdvPickerItem)? = nil)
```

Parameters:

- header: Text for the table heading.
- field: Data field of the table.
- size: Width of the table column.
- hasFocus: Determines whether this table column receives focus when switching to editing mode.
- subType: Sets the data type of the table column.
- pickerColumn: Determines whether it is a drop-down/picker item.

Example Usage

```
AdvTableView(items: $items, selectedItems: $selectedItems, viewModel: vm) {

    AdvTableColumn(header: "lblTableTaxDescriptionShort", field:
      \ArticleTax.descriptionShort, size: .flexible(minimum: 100, maximum: 200),
      hasFocus: true)

    AdvTableColumn(header: "lblTableTaxDescriptionLong",
      field: \ArticleTax.descriptionLong, size: .flexible(minimum: 100))

    AdvTableColumn(header: "lblTableTaxRate", field: \ArticleTax.taxRate,
      subType: .percent)

    AdvTableColumn(header: "lblTableTaxAccount",
      field: \ArticleTax.accountingAccount)

    AdvTableColumn(header: "lblTableTaxValidFrom",
      field: \ArticleTax.validFrom, subType: .date(style: .long, datePicker: true))
}
```