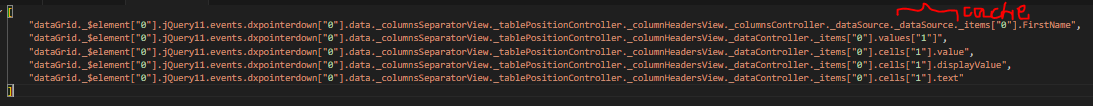
## DataGrid

* DataGrid is a UI component.
* Represent the data from local or remote source in the form of a **grid**.
* Offers basic features like, sorting, grouping, searching.
* Adv. Capabilities like, state storing, client-side exporting, master-detail interface…

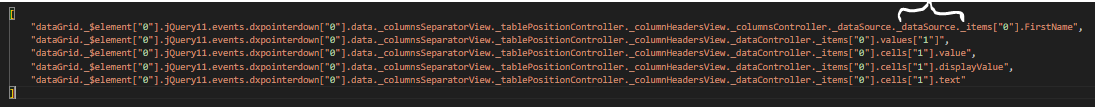
### DataGrid options

* accessKey – no access key in html found
* activeStateEnabled – no visual diff
* allowColumnReordering – user can reorder columns
* allowColumnResizing – user can resize column width
* autoNavigateToFocusedRow – when true, and when focusedRowKey is changed => Automatically scrolls to the focused row (incompatible with **infinite scrolling**).
* cacheEnabled – when performing operations like sorting, grouping, paging => the UI component takes data from this cache. consider disabling it for frequently changing data sources.

Cache enabled:



Cache disabled:



* cellHintEnabled: when cells value is overflowed then enbaling this option, when hover on cell will show complete value of that cell.
* columnAutoWidth: When this property is set to true, all columns adjust their width to the content. columnAutoWidth vs columnWidth: “auto”.
* columnChooser: used to specify which column to hide at runtime. (type – object)
  + allowSeach – seaches column to hide in column chooser box.
  + Empty panel text – text that is to be appear when no column is selected to hide.
  + Enabled – enables column chooser (default f)
  + Heigh and width
  + Mode – specify mode of selecting column to hide (“dragAndDrop”, “select”)
  + searchTimeout
  + title – title of column chooser box
* columnFixing – used to specify which column to fix while horizontal scrolling at runtime. (Type – object)
  + enabled – when true => right clicking on column header gives option in context menu for column fixing (fix and unfix).
  + Texts – object, text to appear in context menu (fix, leftPosition, rightPosition, unfix)
* columnHidingEnabled: should Ui component hide columns to adapt to the screen or container size. If allowColumnResizing is true and columnResizingMode is widget then this property is ignored.
* columnMinWidth: minimum width of the column. (type number).
* columns[]

default: a column is created for each field of DataSource object.

To sepcify which columns should be on ui, u need to use this array.

Each **grid column** is represented by an object containing column setting in this array.

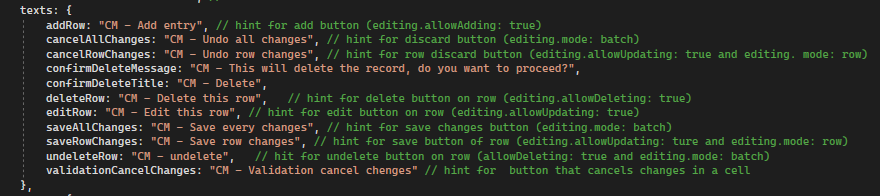
Column properties defines the behaviour and appearance of a grid column.

* alignment – aligns the content of the column. (undefined, “left”, ”right”, “center”)
* allowEditing – can user edit value at runtime. (default inherits from editing.allowEditing). If values in the column are calculated customarily using the calculateCellValue property, they cannot be edited at runtime.
* allowExporting – whether data from this column should be exported. Column should be visible. Do not disable the allowExporting property if you export the DataGrid with multi-line headers.
* allowFiltering – whether data can be filtered by this column.
* allowFixing – whether user can fix the column at runtime. columnFixing should be true.
* allowGrouping – whether the user can group data **by value of this column**. Grouping must be enabled. In a column with calculated values, this property is set to false by default.
* **editing{}**: data grid allow user to add, update, delete data. (allow adding,updating, deleting)
* **allowAdding**: User can add row? (bool)
* **allowDeleting**: user can delete a row? (fn, bool)
* **allowUpdating**: user can update row? (fn, bool)
* **refreshMode**: string

The refreshMode option in DevExtreme's DataGrid specifies the operations that are performed after saving changes. Here's a breakdown of each mode:

1. **Full**:
   * **Data Reloading:** Data is reloaded from the data source. This means that the entire dataset is fetched again, including the data that was just modified, to reflect the latest changes.
   * **Data Processing Operations:** All data processing operations, such as sorting, grouping, and filtering, are reapplied to the reloaded dataset.
   * **UI Component Repaint:** The entire UI component is redrawn to display the reloaded and processed data. This can include re-rendering all rows, headers, and other UI elements.
2. **Reshape**:
   * **Data Reloading: No data reloading occurs.** The current dataset remains the same.
   * **Data Processing Operations:** Data processing operations are reapplied to the current dataset. This means that sorting, grouping, and filtering are performed on the existing data.
   * **UI Component Repaint:** Only the UI component is redrawn to reflect the changes in the processed data. This means that rows and other UI elements may be updated without reloading the entire dataset.
3. **Repaint**:
   * **Data Reloading**: No data reloading occurs. The current dataset remains the same.
   * **Data Processing Operations**: No data processing operations are performed. The current dataset is used as is.
   * **UI Component Repaint**: Only the UI component is redrawn. This means that the UI elements are updated without any changes to the data or its processing.

* **texts**: Contains properties that specify texts for editing-related UI elements.



* **grouping**: grouping allow to **group rows** that share **common values** in one or more column, => making it more easier to analyze and understand data.
* Filtering:

The columns[].filterType property in DevExtreme's DataGrid controls how the header filter behaves when a user interacts with it. It determines whether a user changes the current filter by including (selecting) or excluding (clearing the selection of) values in the filter.

Example Scenario:

Let's say you have a column in your DataGrid that contains the following values: "A", "B", "C", "D", and "E".

Include (Default):

Initially, none of the values are selected in the header filter.

If the user selects "A" and "B", the filter includes only rows where the column value is "A" or "B".

If the user then selects "C", the filter includes rows where the column value is "A", "B", or "C".

Exclude:

Initially, all values are selected in the header filter.

If the user deselects "A" and "B", the filter excludes rows where the column value is "A" or "B".

If the user then deselects "C", the filter excludes rows where the column value is "A", "B", or "C".

**Include**: Users start with no values selected and add values to filter.

**Exclude**: Users start with all values selected and remove values to filter.

* Sorting

Columns[]:

**allowSorting** (true) – determine does this column support sorting

**caculateSortValue** (str, fn) - The calculateSortValue property is used when you want the DataGrid to sort a column based on a value that is not directly displayed in the column. This is useful when the displayed value needs to be sorted differently from its appearance or when sorting should be based on derived or computed values. Used to define custom sorting for individual rows within a column.