**DataGrid**

**Columns:**

a column is created for each field of a data source object, but in most cases, it is redundant. To specify a set of columns to be created in a grid, assign an array specifying these columns to the columns property.

**columnAutoWidth:**

Specifies whether columns should adjust their widths to the content.

Type:bool

Default Value: false

**columnChooser:**

The column chooser allows a user to hide columns at runtime. To enable it, assign true to the columnChooser.[enabled](https://js.devexpress.com/jQuery/Documentation/21_1/ApiReference/UI_Components/dxDataGrid/Configuration/columnChooser/#enabled) property.

allowSearch

emptyPanelText

enabled

height

mode

searchTimeout

title

width

**columnFixing:**fix specific column

enabled

**columnFixing.texts:**

fix

leftPosition

rightPosition

unfix

**Columns[]:**

**alignment**

Aligns the content of the column.

**allowEditing**

Specifies whether a user can edit values in the column at runtime

**allowExporting**

Specifies whether data from this column should be exported

**allowFiltering**

Specifies whether data can be filtered by this column.

**allowFixing**

Specifies whether a user can fix the column at runtime

**allowGrouping**

Specifies whether the user can group data by values of this column. Applies only when grouping is enabled.

**allowHeaderFiltering,** **allowReordering, allowResizing**

**allowSearch , allowSorting ,**

[**buttons[]**](https://js.devexpress.com/jQuery/Documentation/21_1/ApiReference/UI_Components/dxDataGrid/Configuration/columns/buttons/) **:**  custom button

**calculateCellValue , calculateDisplayValue, caption , columns.columns (**nested columns**), customizeText, dataField,**

**datatype, falseText , filterOperations , filterType(**include,exclude**), filterValue ,filterValues , fixed , fixedPosition , format , formItem , headerFilter , hidingPriority , name , selectedFilterOperation , trueText , type , columnWidth**

**and etc.**

**State Persistence:**

**->**State storing enables the UI component to save applied settings and restore them the next time the UI component is loaded. Assign true to the enables:true property to enable this functionality.

**customLoad:**Specifies a function that is executed on state loading. Applies only if the type is *'custom'*.

**customSave:** Specifies a function that is executed on state change. Applies only if the type is *"custom"*.

**Enabled : true , false**

**savingTimeout:** delay in millisecond to changes in store(local,session,custom)

**storageKey:** specify key in storage

**type :** Specifies the type of storage where the state is saved.

**Template:**

a template is a mechanism that allows you to customize the way UI components display content.

**Purpose:** Customizes how data is displayed within a column in components like the **DataGrid**.

**Use Case:** You can create a custom layout or style for the column's content, such as embedding images, custom text formatting, or additional UI components.

|  |
| --- |
| Column Template |
| Row Template |
| Cell Customization |
| Toolbar Customization |

**Summaries**

**Data Summaries:**

* These are summaries that are calculated directly from the data itself, such as sum, average, count, or other aggregate functions.
* Typically used in DataGrid components to display calculated values for columns or rows.

**Group Summaries:**

* Group summaries are used when data is grouped in the grid (via grouping rows by a specific field).
* These summaries provide aggregation values for each group (such as sum, average, etc.) rather than the entire dataset.

**Custom Summaries:**

* Custom summaries allow you to define your own summary calculation logic.
* You can implement a custom function to summarize data, such as applying business-specific rules, formulas, or custom aggregations.

**Master-Detail:**

a master-detail interface supplies a usual data row with an expandable section that contains the details on this data row. In that case, the data row is called "master row", while the section is called "detail section".

**autoExpandAll**

**enabled**

**template**

**Export:**

Configures client-side exporting.

A user can click the Export button to save an Excel file with the exported data. Data types, sort, filter, and group settings are maintained.

**First:** Excel sheet

**Second :** pdf

**Adaptability**