**DataGrid**

**DataSource:**

**ArrayStore**: Stores data locally in memory, no backend required.

**CustomStore**: Provides custom logic for working with remote or custom data sources.

**DataSource**: An abstraction layer for managing data from any store, supporting powerful features like filtering, sorting, and paging.

**LocalStore**: A store that uses browser (localStorage/sessionStorage) for persisting small datasets on the client side.

**Query**: A query abstraction used to build and perform data operations like filtering and sorting on any data source.

**Data Binding:**

**Simple Array:** Uses a local JavaScript array as the data source.

**Ajax Request:**Fetches data from an API or server using AJAX.

**Paging:**

Paging allows the UI component to load data in portions instead of loading it simultaneously. To enable paging, set the paging [enabled](https://js.devexpress.com/jQuery/Documentation/21_1/ApiReference/UI_Components/dxDataGrid/Configuration/paging/#enabled) property to true.

Users can switch between pages and change paging settings using the [pager](https://js.devexpress.com/jQuery/Documentation/21_1/ApiReference/UI_Components/dxDataGrid/Configuration/pager/) or they can scroll the pages. Paging settings apply with any [scrolling mode](https://js.devexpress.com/jQuery/Documentation/21_1/ApiReference/UI_Components/dxDataGrid/Configuration/scrolling/#mode).

**Pager:**

The pager is an element that allows users to navigate through pages and change their size at runtime.

The pager consists of the page navigator and several optional elements:

page size selector, navigation buttons, and page information.

**Editing:**

Row Editing & Events – Edits entire rows with save/cancel actions.

Property: editing.mode – Defines editing mode ("row" for row editing).

Method: editRow(rowIndex) – Puts a specific row into edit mode.

Event: onRowUpdating – Triggered before a row update.

**Batch Editing** – Allows multiple edits before saving changes.

**Mode**: editing.mode – Set to "batch" for batch editing.

**Cell Editing** – Edits individual cells without affecting the row.

**Mode**: editing.mode – Set to "cell" for cell editing.

**Form Editing** – Opens a form to edit row data.

**Mode**: editing.mode – Set to "form" for form-based editing.

**Popup Editing** – Edits rows in a modal popup.

**Mode**: editing.mode – Set to "popup" for popup editing.

**Data Validation** – Ensures input follows rules (e.g., required, max length).

**Property**: columns.validationRules – Defines validation rules for a column.

**Method**: validate() – Validates all fields.

**Event**: onRowValidating – Triggered before row validation.

**Cascading Lookups** – Dynamically filters lookup fields based on selection.

**Property**: lookup.dataSource – Defines the lookup data source.

**Event**: onEditorPreparing – Adjusts lookup options dynamically.

**Grouping:**

grouping is a feature that allows you to organize data based on specific column values.

allowCollapsing

autoExpandAll

autoExpandAll

expandMode

**GroupPanel:**

Data in DataGrid can be grouped by one column or by several. Once a column is used for grouping, it is added to the group panel.

By default, the group panel is hidden.

To make it visible, set the groupPanel [visible](https://js.devexpress.com/jQuery/Documentation/21_1/ApiReference/UI_Components/dxDataGrid/Configuration/groupPanel/#visible) property to **true**.

**Filtering:**

The filter row allows a user to filter data by individual columns' values.

Column-Based Filtering

Multiple Filter Conditions

Please filter panel must be visible true.

**Sorting:**

sorting is a feature that allows users to arrange data in a specific order based on one or more columns.

Key Features of Sorting:

Column-based Sorting, Sorting Index

Sorting Options

Sorting Modes: ASC,DESC

**Selection:**

Configures runtime selection.

A user can select rows in a single or multiple [mode](https://js.devexpress.com/jQuery/Documentation/21_1/ApiReference/UI_Components/dxDataGrid/Configuration/selection/#mode). In multiple mode, a user can select all rows at once. To disable this feature, assign false to the [allowSelectAll](https://js.devexpress.com/jQuery/Documentation/21_1/ApiReference/UI_Components/dxDataGrid/Configuration/selection/" \l "allowSelectAll).

**allowSelectAll**

**deferred**

**mode**

**selectAllMode**

**showCheckBoxesMode**

**Sorting:**

A user can sort rows by values of a single or multiple columns depending on the value of the sorting.[mode](https://js.devexpress.com/jQuery/Documentation/21_1/ApiReference/UI_Components/dxDataGrid/Configuration/sorting/#mode) property.

**ascendingText**

**clearText**

**descendingText**

**mode**

**showSortIndexes**

**filterPanel:**

The filter panel displays the applied filter expression.

**customizeText**

**filterEnabled**

[**texts**](https://js.devexpress.com/jQuery/Documentation/21_1/ApiReference/UI_Components/dxDataGrid/Configuration/filterPanel/texts/)

**visible**

**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:**

* columnHidingEnabled → Enables automatic column hiding.
* adaptivityEnabled → Enables adaptive layout for grids.