# React DataGrid Datasheet

## Overview

The Syncfusion React Grid is a feature-rich and high-performing component useful for displaying data in a tabular format. Its wide range of functionalities includes data binding, adaptive UI layout for all devices, editing, Excel-like filtering, custom sorting, aggregating rows, selection, and support for Excel, CSV, and PDF formats. It is compatible with Next.JS, Preact, and Gatsby.

## Key features

* **Data Sources**: Easily bind data to the grid from local (JavaScript arrays) or remote (RESTful JSON) sources, allowing you to handle large datasets or fetch data remotely.
* **Adaptors**: Connect to different backend services and APIs effortlessly, making it easier to integrate Syncfusion components with your data sources.
* **Adaptive UI Layout**: The grid's UI adapts for optimal viewing on small screens, rendering features like filters, sorting, and edit dialogs effectively. You can also display grid rows vertically for better readability.
* **Sorting and Grouping**: Sort data in ascending or descending order and group records hierarchically. Expand and collapse records with ease, and apply multiple levels of sorting and grouping.
* **Filtering**: Filter data based on specific criteria to focus on what's important, making data analysis and decision-making easier.
* **Paging**: Display grid data in pages, making it easier to navigate through large datasets.
* **Editing**: Add, remove, and update records directly within the grid, supporting seamless CRUD (Create, Read, Update, Delete) operations.
* **Columns**:
  + **Reordering**: Drag and drop column headers to reorder columns.
  + **Chooser**: Toggle column visibility with a column chooser.
  + **Resizing**: Easily resize columns to fit their content.
  + **Frozen Rows and Columns**: Keep critical information visible while scrolling through large datasets.
  + **Spanning**: Merge adjacent cells horizontally or vertically to create a cleaner layout.
  + **Foreign Key Binding**: Display related data from a foreign key source.
  + **Cell Styling**: Customize cell appearance with fonts, colors, and styles.
* **Selection**: Select specific cells, rows, or columns using mouse clicks, keyboard arrows, or touch for highlighting and performing actions on them.
* **Templates**: Use custom elements like images or buttons in a column instead of the field value.
* **Aggregation**: Show aggregate values like sums or averages in the footer or group captions for quick calculations.
* **Context Menu**: Right-click to access a context menu for quick actions on grid data.
* **Clipboard**: Copy selected rows or cells to the clipboard using keyboard shortcuts.
* **Export**: Export grid data to Excel, CSV, or PDF formats for easy sharing and analysis.
* **Mobile View**: Mobile-first design that adapts to any resolution, ensuring an optimal experience on both mobile and desktop devices.
* **Accessibility**: Complies with accessibility standards like ADA, Section 508, and WCAG 2.2 to ensure usability for all users.
* **RTL (Right-to-Left)**: Switch text direction and layout for languages like Arabic, Farsi, Urdu and more.
* **Localization**: Customize grid text to support different languages and cultural preferences with a built-in localization library.

## Resources

[Documentation](https://ej2.syncfusion.com/react/documentation/grid/getting-started) – [API Docs](https://ej2.syncfusion.com/react/documentation/api/grid/) - [Video Tutorials](https://www.syncfusion.com/tutorial-videos/react/grid) – [Blog articles](https://www.syncfusion.com/blogs/?s=react+grid) – [Online demos](https://ej2.syncfusion.com/react/demos/#/fluent2/grid/overview) – [Use case Examples](https://github.com/SyncfusionExamples/react-grid-use-case-tutorial-samples)

## Performance metrics

Syncfusion React DataGrid excels in performance, especially with large datasets. Below are the metrics for a grid with 100,000 records and 5 columns, demonstrating the efficiency of paging and virtualization. You can test these examples from our GitHub repository: <https://github.com/SyncfusionExamples/React-Grid-performance-sample>

For tips and best practices on improving the performance of the Syncfusion Data Grid, refer to our documentation: [Improving Syncfusion Data Grid Performance](https://ej2.syncfusion.com/react/documentation/grid/performance).

**Local Data:**

**Paging Performance Metrics**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Page Size** | **Initial Load** | **Paging** | **Filtering** | **Clear Filtering** | **Sorting (with 2 columns)** | **Grouping (with 2 columns)** |
| 100 | ~0.037 secs | ~0.032 secs | ~0.069 secs | ~0.038 secs | ~1.515 secs | ~1.794 secs |
| 1000 | `~0.127 secs | ~0.108 secs | ~0.149 secs | ~0.117 secs | ~1.686 secs | ~1.899 secs |
| 2000 | ~0.203 secs | ~0.210 secs | ~0.223 secs | ~0.203 secs | ~1.921 secs | ~2.139 secs |

**Virtualization Performance Metrics**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Page Size** | **Initial Load** | **Filtering** | **Clear Filtering** | **Sorting (with 2 columns)** | **Grouping (with 1 column)** |
| 100 | ~0.052 secs | ~0.071 secs | ~0.037 secs | ~1.771 secs | ~2.319 secs |
| 1000 | ~0.249 secs | ~0.176 secs | ~0.134 secs | ~1.866 secs | ~2.596 secs |
| 2000 | ~0.467 secs | ~0.246 secs | ~0.200 secs | ~1.972 secs | ~2.906 secs |

**Remote data (URL adaptor)**

**Paging Performance Metrics**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Page Size** | **Initial Load** | **Paging** | **Filtering** | **Clear Filtering** | **Sorting (with 2 columns)** | **Grouping (with 2 columns)** |
| 100 | ~0.463 secs | ~0.454 secs | ~0.556 secs | ~0.424 secs | ~0.912 secs | ~1.046 secs |
| 1000 | ~0.598 secs | ~0.610 secs | ~0.842 secs | ~0.580 secs | ~1.086 secs | ~1.488 secs |
| 2000 | ~0.912 secs | ~0.902 secs | ~1.191 secs | ~0.956 secs | ~1.352 secs | ~2.234 secs |

**Virtualization Performance Metrics**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Page Size** | **Initial Load** | **Filtering** | **Clear Filtering** | **Sorting (with 2 columns)** | **Grouping (with 1 column)** |
| 100 | ~0.400 secs | ~0.576 secs | ~0.380 secs | ~0.890 secs | ~1.010 secs |
| 1000 | ~0.814 secs | ~0.880 secs | ~0.616 secs | ~1.098 secs | ~1.758 secs |
| 2000 | ~1.182 secs | ~1.178 secs | ~0.902 secs | ~1.412 secs | ~2.674 secs |

## Integrating React Grid with Frameworks and Tools

Our React Grid has been successfully integrated into the following frameworks and tools:

|  |  |
| --- | --- |
| **Frameworks:**   * SharePoint * Preact * Next.js * Gatsby | **Build Tools and Applications:**   * Vite (Build tool for faster development and builds) * Remix (Full-stack web framework) * Webpack Eternals (Optimization feature in Webpack) |

## Supported React Version

The table below lists the supported React versions for various Syncfusion React UI component releases.

| **Version** | **Syncfusion React components version** |
| --- | --- |
| [React v18.0](https://reactjs.org/blog/2022/03/29/react-v18.html) | 20.2.36 and above |
| [React v17.0](https://reactjs.org/blog/2020/10/20/react-v17.html) | 18.3.50 and above |
| [React v16.0](https://reactjs.org/blog/2017/09/26/react-v16.0.html) | 16.2.45 and above |

## Browser Support

The Syncfusion React components are supported in all modern browsers. This includes the following versions.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Chrome | Firefox | Opera | Edge | IE | Safari | IOS | Android | Windows Mobile |
| Latest | Latest | Latest | 13 + | 11 + | 9 + | 9 + | 4.4 + | IE 11 + |

## Why choose Syncfusion Data Grid

* High Performance
* Support for multiple frameworks such as JavaScript, React, Angular, Vue, ASP.NET Core with the same API.
* Provides a complete set of controls for your application such as Charts, Pivot Table, Tree Grid, Scheduler, Gantt Chart, and more.
* Paging – Responsive pager, Custom paging, Dynamic page size changes
* Virtualization – infinite scrolling with cache and column virtualization
* Exporting – PDF, Image column, customization
* Mobile vertical layout
* Toggle column visibility with a column chooser UI

## General questions: