



Quick Start

This page covers the key concepts of AG Grid: A high performance, feature rich, **React Table** and React Data Grid.



Your First React Data Grid

Add AG Grid to your application in these steps:

1. NPM Install

```
npm install ag-grid-react
```

2. Import the React Data Grid

```
import { AgGridReact } from 'ag-grid-react'; // React Data Grid Component
import "ag-grid-community/styles/ag-grid.css"; // Mandatory CSS required by the L
import "ag-grid-community/styles/ag-theme-quartz.css"; // Optional Theme applied
```

3. Define Rows and Columns

```
const GridExample = () => {
  // Row Data: The data to be displayed.
  const [rowData, setRowData] = useState([
    { id: 1, name: 'John', age: 30, salary: 10000 },
    { id: 2, name: 'Jane', age: 25, salary: 12000 },
    { id: 3, name: 'Mike', age: 35, salary: 15000 },
    { id: 4, name: 'Sarah', age: 28, salary: 18000 },
    { id: 5, name: 'David', age: 32, salary: 20000 },
    { id: 6, name: 'Emily', age: 27, salary: 22000 },
    { id: 7, name: 'Chris', age: 31, salary: 25000 },
    { id: 8, name: 'Alex', age: 29, salary: 28000 },
    { id: 9, name: 'Lisa', age: 33, salary: 30000 },
    { id: 10, name: 'Tom', age: 34, salary: 32000 },
    { id: 11, name: 'Mia', age: 26, salary: 35000 },
    { id: 12, name: 'Noah', age: 36, salary: 38000 },
    { id: 13, name: 'Sophia', age: 24, salary: 40000 },
    { id: 14, name: 'Liam', age: 37, salary: 42000 },
    { id: 15, name: 'Olivia', age: 23, salary: 45000 },
    { id: 16, name: 'Ethan', age: 38, salary: 48000 },
    { id: 17, name: 'Ava', age: 22, salary: 50000 },
    { id: 18, name: 'Caleb', age: 39, salary: 52000 },
    { id: 19, name: 'Mia', age: 21, salary: 55000 },
    { id: 20, name: 'Ethan', age: 40, salary: 58000 },
  ])
}
```

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

[Do Not Sell My
Personal Information](#)

Accept Cookies

```

    { field: "model" },
    { field: "price" },
    { field: "electric" }
  ]);

  // ...

}

```

4. React Data Grid Component

The `AgGridReact` component is wrapped in a parent container `div`. Style is applied to the parent container. Rows and Columns are set as `AgGridReact` component attributes.

```

return (
  // wrapping container with theme & size
  <div
    className="ag-theme-quartz" // applying the Data Grid theme
    style={{ height: 500 }} // the Data Grid will fill the size of the parent container
  >
    <AgGridReact
      rowData={rowData}
      columnDefs={colDefs}
    />
  </div>
)

```

5. Running the React Data Grid

Below is a live example of the application running. Click `</> Code` to see the code.

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

[Do Not Sell My
Personal Information](#)

Accept Cookies



To live-edit the code, open the example in [CodeSandbox](#) or [Plunker](#) using the buttons to the lower-right.

Now that you have a basic React Data Grid running, the remainder of this page explores some of the key concepts.

Showing Data

Mapping Values

The `field` or `valueGetter` attributes **Map Data to Columns**. A field maps to a field in the data. A **Value Getter** is a function callback that returns the cell value.

The `headerName` provides the title for the header. If missing the title is derived from `field`.

```
const [columnDefs, setColumnDefs] = useState([
  { headerName: "Make & Model", valueGetter: p => p.make + ' ' + p.model },
  { field: "price" },
]);

<AgGridReact columnDefs={columnDefs} />
```

Text Formatting

Format text for cell content using a **Value Formatter**.

```
const [columnDefs, setColumnDefs] = useState([
  { field: "price", valueFormatter: p => '£' + p.value.toLocaleString() },
```

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

**[Do Not Sell My
Personal Information](#)**

Accept Cookies

```
const CustomButtonComponent = (props) => {
  return <button onClick={() => window.alert('clicked')} >Push Me!</button>;
};

const [colDefs, setColDefs] = useState([
  { field: "button", cellRenderer: CustomButtonComponent },
  // ...
]);
```

Resizing Columns

Columns are **Resized** by dragging the Column Header edges. Additionally assign `flex` values to allow columns to flex to the grid width.

```
const [columnDefs, setColumnDefs] = useState([
  { field: "make", flex: 2 }, //This column will be twice as wide as the others
  { field: "model", flex: 1 },
  { field: "price", flex: 1 },
  { field: "electric", flex: 1 }
]);

<AgGridReact columnDefs={columnDefs} />
```

Example

This example demonstrates mapping and formatting values, cell components, and resizing columns.

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

[Do Not Sell My
Personal Information](#)

Accept Cookies

Working with Data

By default, the row data is used to infer the **Cell Data Type**. The cell data type allows grid features, such as filtering and editing, to work without additional configuration.

Filtering

Column Filters are embedded into each column menu. These are enabled using the `filter` attribute. The filter type is inferred from the cell data type.

```
const [columnDefs, setColumnDefs] = useState([
  { field: "make", filter: true },
]);
```

```
<AgGridReact columnDefs={columnDefs} />
```

There are 5 **Provided Filters** which can be set through this attribute. You can also create your own **Custom Filter**.

Floating Filters embed the Column Filter into the header for ease of access.

```
const [columnDefs, setColumnDefs] = useState([
  { field: "make", filter: true, floatingFilter: true },
]);
```

```
<AgGridReact columnDefs={columnDefs} />
```

Editing

Enable **Editing** by setting the `editable` attribute to `true`. The cell editor is inferred from

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

**[Do Not Sell My
Personal Information](#)**

Accept Cookies

Set the cell editor type using the `cellEditor` attribute. There are 7 **Provided Cell Editors** which can be set through this attribute. You can also create your own **Custom Editors**.

```
const [columnDefs, setColumnDefs] = useState([
  {
    field: "make",
    editable: true,
    cellEditor: 'agSelectCellEditor',
    cellEditorParams: {
      values: ['Tesla', 'Ford', 'Toyota'],
    },
  },
]);

<AgGridReact columnDefs={columnDefs} />
```

Sorting

Data is **Sorted** by clicking the column headers. Sorting is enabled by default.

Row Selection

Row Selection is enabled using the `rowSelection` attribute.

```
// Column Definitions: Defines the columns to be displayed.
const [columnDefs, setColumnDefs] = useState([
  { field: "make" },
]);
const rowSelection = useMemo(() => {
  return {
    mode: 'multiRow',
  };
}, []);
```

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

**[Do Not Sell My
Personal Information](#)**

Accept Cookies

Pagination

Enable **Pagination** by setting `pagination` to be true.

```
const pagination = true;
const paginationPageSize = 500;
const paginationPageSizeSelector = [200, 500, 1000];

<AgGridReact
  pagination={pagination}
  paginationPageSize={paginationPageSize}
  paginationPageSizeSelector={paginationPageSizeSelector}
/>
```

Example

This example demonstrates filtering, editing, sorting, row selection, and pagination.



This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

[Do Not Sell My
Personal Information](#)

Accept Cookies

Themes & Style

Themes

Grid Themes define how the grid looks (colors, font, spacing etc). The grid comes with **Provided Themes** such as Quartz and Alpine. To use a theme you need to 1) import the themes CSS and 2) apply the theme to the parent HTML element of the grid.

```
import "ag-grid-community/styles/ag-theme-quartz.css"; // import Quartz theme
// import "ag-grid-community/styles/ag-theme-alpine.css"; // import Alpine theme,
...
return (
  <div class="ag-theme-quartz"> // set Quartz Theme on parent div
    <AgGridReact rowData={...} columnDefs={...} />
  </div>
)
```

The image shows four overlapping grid tables, each representing a different theme. The tables are arranged from left to right, with each subsequent table shifted slightly to the right and top, creating a layered effect. The themes shown are Alpine Light, Alpine Dark, Quartz Light, and Quartz Dark. Each table has two columns: 'Make' and 'Model'. The data rows are: Tesla, Ford, Toyota, Mercedes, Fiat, and Nissan. The Quartz themes (Light and Dark) have a wider column width compared to the Alpine themes (Light and Dark).

Make	Model
Tesla	Model S
Ford	F-Series
Toyota	Corolla
Mercedes	EQA
Fiat	500
Nissan	Juke

From left to right **Alpine Light**, **Alpine Dark**, **Quartz Light**, and **Quartz Dark** AG Grid themes.

Customising a Theme

Customise Themes using CSS variables.

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

[Do Not Sell My
Personal Information](#)

Accept Cookies


```
--ag-header-background-color: rgb(228, 237, 250);  
/* Changes the hover color of the row*/  
--ag-row-hover-color: rgb(216, 226, 255);  
}
```

[</> Code](#)

Figma

If you are designing within Figma, you can use the **AG Grid Design System** to replicate the Quartz and Alpine AG Grid themes within Figma. These default themes can be extended with Figma variables to match any existing visual design or create entirely new AG Grid themes. These can then be exported and generated into new AG Grid themes.

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

[Do Not Sell My
Personal Information](#)

Accept Cookies

to set cell background colour based on its value.

```
.rag-green {  
  background-color: #33cc3344;  
}  
  
const [columnDefs, setColumnDefs] = useState([  
  field: 'electric',  
  cellClassRules: {  
    // apply green to electric cars  
    'rag-green': params => params.value === true,  
  }  
]);  
  
<AgGridReact columnDefs={columnDefs} />
```

Row Style

Define rules to apply **Styling to Rows** using `rowClassRules`. This allows changing style (e.g. row colour) based on row values.

```
.rag-red {  
  background-color: #cc222244;  
}  
  
const rowClassRules = {  
  // apply red to Ford cars  
  'rag-red': params => params.data.make === 'Ford',  
};  
  
<AgGridReact rowClassRules={rowClassRules} />
```



`</>` Code



Next Steps

- Read our [Introductory Tutorial](#).
- Watch our [Video Tutorials](#).

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

[Do Not Sell My
Personal Information](#)

Accept Cookies

Documentation

[Getting Started](#)

[Changelog](#)

[Pipeline](#)

[Documentation Archive](#)

Support & Community

[Stack Overflow](#)

[License & Pricing](#)

[Support via Zendesk](#)

The Company

[AG Charts](#)

[About](#)

[Blog](#)

[Privacy Policy](#)

[Cookies Policy](#)

[Sitemap](#)

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

**[Do Not Sell My
Personal Information](#)**

Accept Cookies

[X ↗](#)

[YouTube ↗](#)

[LinkedIn ↗](#)

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [Cookies Policy](#)

[Do Not Sell My
Personal Information](#)

Accept Cookies