

SciChart Documentation is best viewed inside a Navigation Frame.

Click [HERE](#) to load it!

JavaScript Charting Documentation - SciChart JavaScript Charts SDK v3.x



Get Started: Tutorials, Examples > Tutorials (SciChart React) > Tutorial 04 - Adding & Removing Charts To a Group in React

Tutorial 04 - Adding & Removing Charts To a Group in React

In the previous tutorials we showed you how to use `<SciChartReact/>` onInit and React State / Contexts to allow you to control chart data and chart state via buttons hosted in your app.

In this tutorial, we're going to show how to dynamically add & remove chart panels from a `<SciChartGroup/>` - a React component in scichart-react which allows you to host several charts together in a group, and synchronize zooming, panning and axis sizes in order to create dynamic linked dashboards using SciChart.js.



As a basis for this tutorial, use [Tutorial 01 - Understanding the scichart-react boilerplate](#) for the initial project setup. Go ahead and copy the boilerplate to a new folder or project. You can get the code from here: [Boilerplates/scichart-react](#)

Before we Begin

Let's roll back some changes from the previous tutorial. We want our App.jsx to be simpler, removing `ToggleButton`, `<SciChartReact/>` and the `ChartContext`.

App.jsx

Copy Code

```
import React from "react";
import "./styles.css";
function App() {
  return (
    <div className="App">
```

```

    <header className="App-header">
      <h1>&lt;SciChartReact/&gt; with custom chart controls</h1>
    </header>
    <div
      style={{
        display: "flex",
        justifyContent: "left",
        backgroundColor: "lightgrey",
        padding: "10px",
      }}
    >
      { /* TODO: Add chart controls here */ }
    </div>
    { /* TODO: Add SciChartReact here */ }
  </div>
);
}
export default App;


```

Here we want to create a `<SciChartGroup/>` which can host one or more `<SciChartReact/>` components to make our dashboard or group of charts.

Import `{ SciChartGroup, SciChartReact }` from `"scichart-react"` to begin, and replace the `{ /* TODO: Add SciChartReact here */ }` with a `<SciChartGroup/>` component.

Place two `<SciChartReact/>` components as children to the `<SciChartGroup/>` and for now, create a placeholder `initChart` function just to create a simple chart with x and y axis.

App.jsx

 Copy Code

```

import React from "react";
import "./styles.css";
import { SciChartGroup, SciChartReact } from "scichart-react";
import { SciChartSurface, NumericAxis, SciChartJsNavyTheme } from "scichart";
const simpleChart = async (divElement, chartId) => {
  const { sciChartSurface, wasmContext } = await SciChartSurface.create(
    divElement,
    {
      title: `Chart ${chartId}`,
      titleStyle: { fontSize: 28 },
      theme: new SciChartJsNavyTheme(),
    }
  );
  sciChartSurface.xAxes.add(
    new NumericAxis(wasmContext, { axisTitle: "X Axis" })
  );
  sciChartSurface.yAxes.add(
    new NumericAxis(wasmContext, { axisTitle: "Y Axis" })
  );
  return { sciChartSurface };
};
function App() {

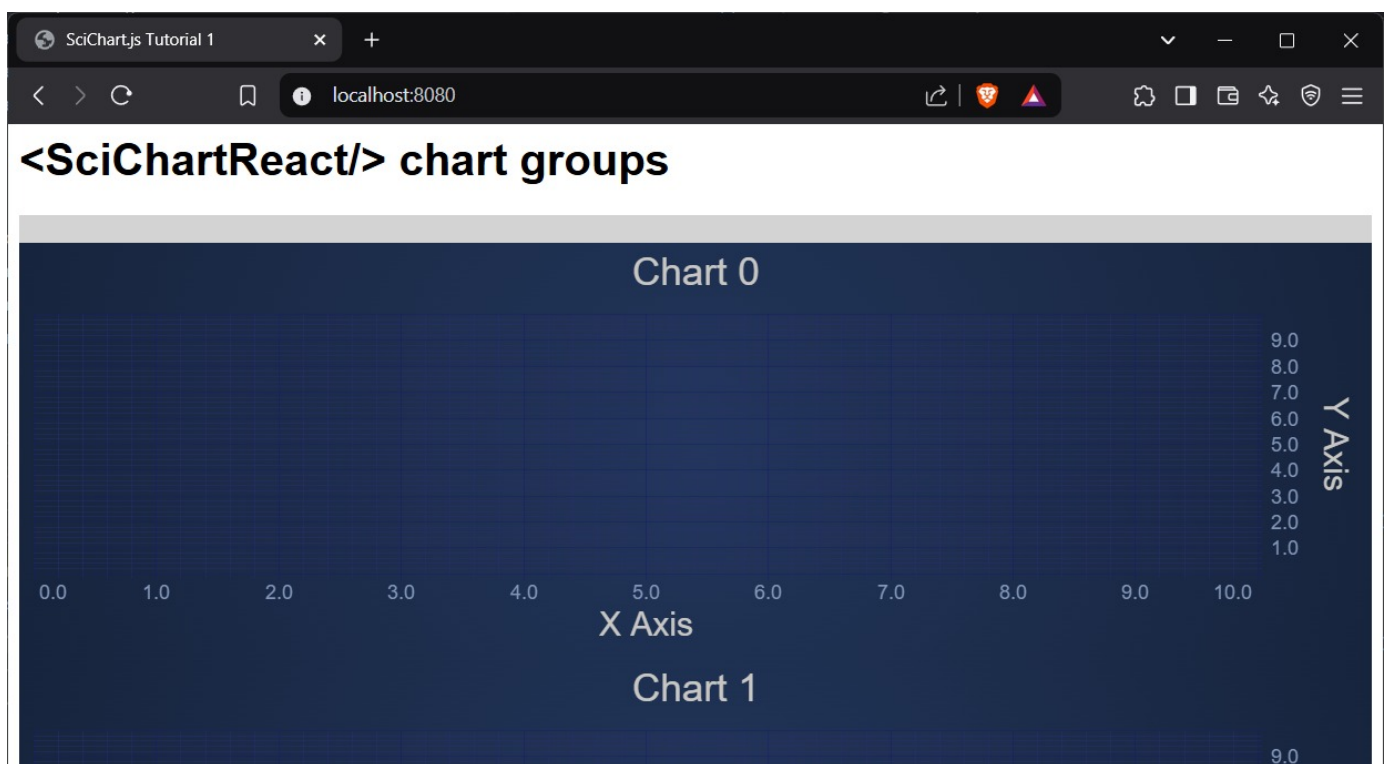
```

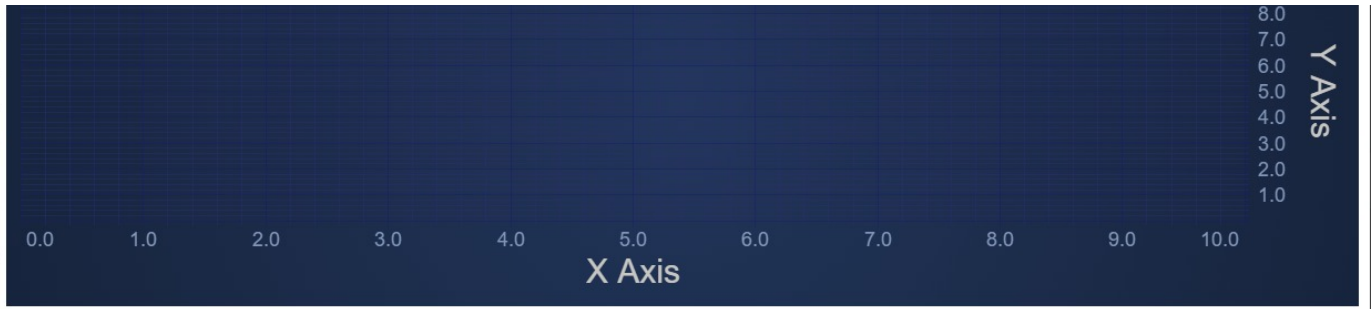
```

return (
  <div className="App">
    <header className="App-header">
      <h1>&lt;SciChartReact/&gt; chart groups</h1>
    </header>
    <div
      style={{
        display: "flex",
        justifyContent: "left",
        backgroundColor: "lightgrey",
        padding: "10px",
      }}
    >
      </* TODO: Add chart controls here */>
    </div>
    <div style={{ height: "600px" }}>
      <SciChartGroup>
        <SciChartReact
          initChart={(div) => simpleChart(div, 0)}
          style={{ height: "50%" }}
        />
        <SciChartReact
          initChart={(div) => simpleChart(div, 1)}
          style={{ height: "50%" }}
        />
      </SciChartGroup>
    </div>
  </div>
);
}
export default App;

```

If you run the app now, you should see the following output:





Note that `<SciChartGroup/>` is a non visual React component, so if you want to set the size of the entire chart group, consider wrapping it in a `<div>` and setting the style property. Then, individual `<SciChartReact/>` components can have relative sizes such as "50%"

Dynamically Adding/Removing Charts to `<SciChartGroup/>`

Next, want to make this example dynamic, by adding a toolbar button to Add and Remove a chart. We will do this by adding/removing `<SciChartReact/>` components directly from the `<SciChartGroup/>` component.

Start by importing `useState` from "react"

App.jsx

[Copy Code](#)

```
import React, { useState } from "react";
```

Next, we want to add an initial state with our charts. This will simply be an array of numbers but it could be an array of objects storing chart related data, configuration or parameters. We want to add an `addChart` and `removeChart` function which manipulates the array.

App.jsx

[Copy Code](#)

```
function App() {
  const [charts, setCharts] = useState([0, 1]); // Initialize with 2 charts
  const addChart = () => {
    setCharts([...charts, charts.length]);
  };
  const removeChart = () => {
    if (charts.length > 0) {
      setCharts(charts.slice(0, -1));
    }
  };
};
```

Finally, update the App component to return buttons to add and remove a chart. We will update the children of `<SciChartGroup/>` to map the `charts` state to 0..N `<SciChartReact/>` components.

The height of each `<SciChartReact/>` component is dynamically adjusted based on the number of charts, to ensure the total parent group size does not exceed 600px.

Here's the final App.jsx source code:

APP.JSX

```
import React, { useState } from "react";
import "./styles.css";
import { SciChartGroup, SciChartReact } from "scichart-react";
import { SciChartSurface, NumericAxis, SciChartJsNavyTheme } from "scichart";

const simpleChart = async (divElement, chartId) => {
  const { sciChartSurface, wasmContext } = await SciChartSurface.create(
    divElement,
    {
      title: `Chart ${chartId}`,
      titleStyle: { fontSize: 16 },
      theme: new SciChartJsNavyTheme(),
    }
  );
  sciChartSurface.xAxes.add(
    new NumericAxis(wasmContext, {
      axisTitle: "X Axis",
      axisTitleStyle: { fontSize: 12 },
    })
  );
  sciChartSurface.yAxes.add(
    new NumericAxis(wasmContext, {
      axisTitle: "Y Axis",
      axisTitleStyle: { fontSize: 12 },
    })
  );

  return { sciChartSurface };
};

function App() {
  const [charts, setCharts] = useState([0, 1]); // Initialize with 2 charts

  const addChart = () => {
    setCharts([...charts, charts.length]);
  };

  const removeChart = () => {
    if (charts.length > 0) {
      setCharts(charts.slice(0, -1));
    }
  }
}
```

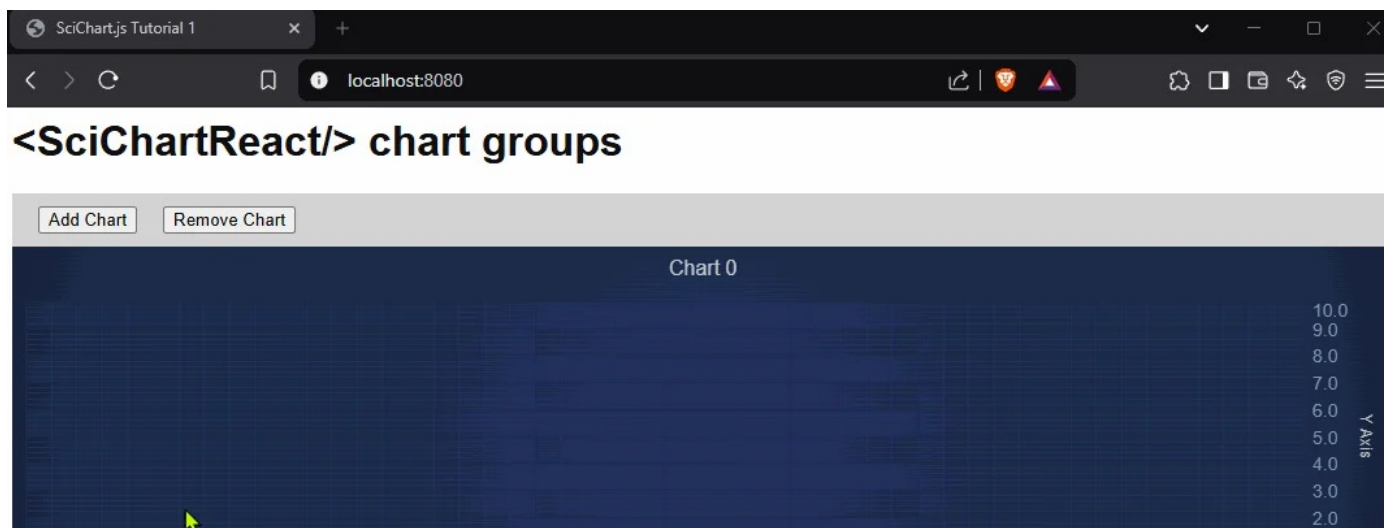
```

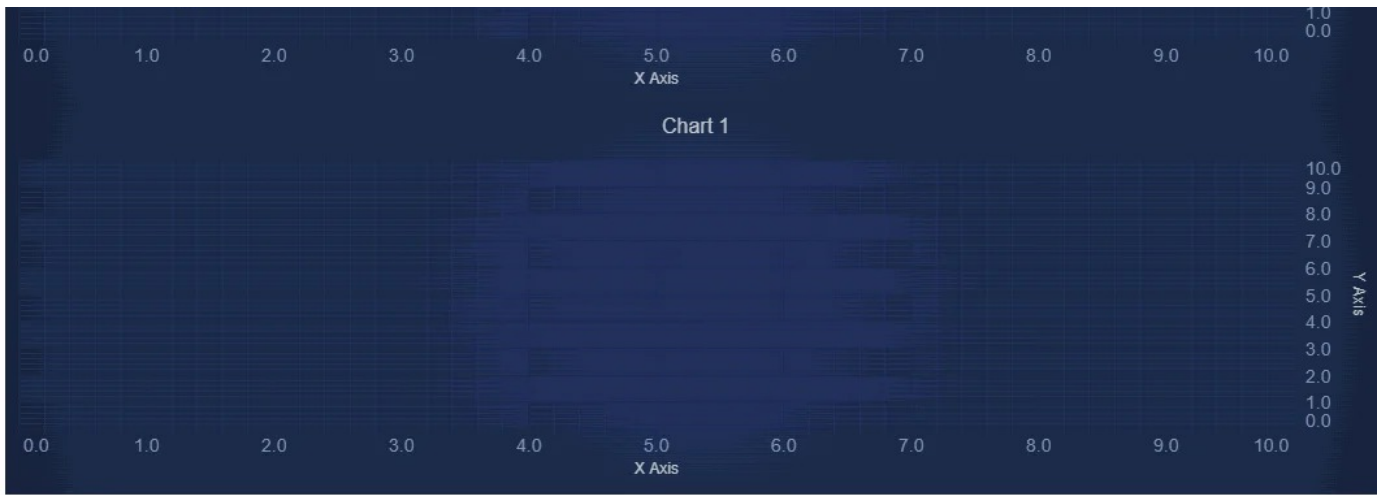
};
return (
  <div className="App">
    <header className="App-header">
      <h1>&lt;SciChartReact&gt; chart groups</h1>
    </header>
    <div
      style={{
        display: "flex",
        justifyContent: "left",
        backgroundColor: "lightgrey",
        padding: "10px",
      }}
    >
      <button onClick={addChart} style={{ margin: "0 10px" }}>
        Add Chart
      </button>
      <button onClick={removeChart} style={{ margin: "0 10px" }}>
        Remove Chart
      </button>
    </div>
    <div style={{ height: "600px" }}>
      <SciChartGroup>
        {charts.map((chartId) => (
          <SciChartReact
            key={chartId}
            initChart={(div) => simpleChart(div, chartId)}
            style={{ height: `${100 / charts.length}%` }}
          />
        ))}
      </SciChartGroup>
    </div>
  </div>
);
}

export default App;

```

Now go ahead and run the app, you should see this:





You can get the full source code for this tutorial over at scichart.js.examples on Github under the folder [Tutorials/React/Tutorial_04_Adding_Removing_Syncing_Charts](#)

SCICHART ® is a Registered Trademark in the UK, US and EU. Copyright SciChart Ltd 2011-2024.

[Sitemap](#) | [Send Feedback](#) 