

Setting up and using Chart Js

This file documents the basic process in setting up the Chart.js library, which allows for the user to access graphing and charting functionalities. It presumes that React.js has already been installed within the 'App's' file structure.

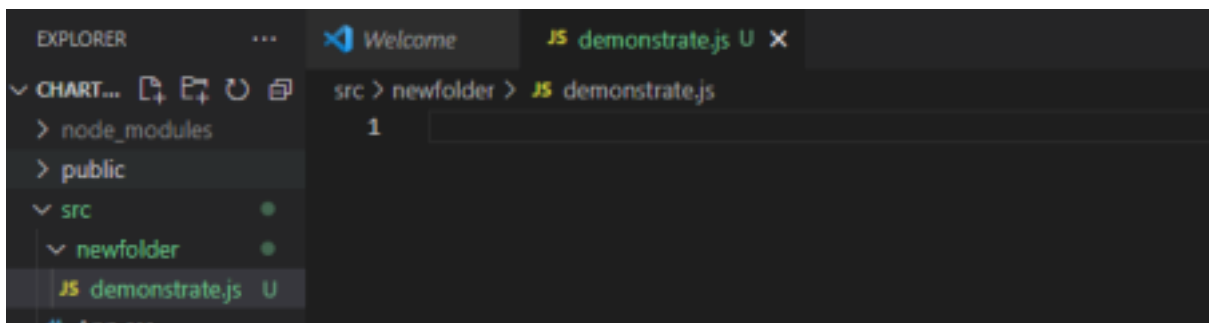
1. Opening a new terminal copy and paste the below command then run it. This will install all associated peer dependencies needed for the Chart.js library and the library itself. This concludes the setup process.

```
npm install --save react-chartjs-2 chart.js
```

The above code can be accessed from the link:

<https://www.npmjs.com/package/react-chartjs-2>

2. To demonstrate how to link a chart to our primary App, we will create a new folder within the 'src' folder. Within this folder create a new .js file to hold the chart/graph functionalities.



3. Copy and paste the below two lines of code. The first will import the necessary React node modules to utilize react. The second imports the chartjs functionalities. Note that 'Line' in this case specifies a line graph, it may be interchanged for different specifiers such as 'Bar', 'Doughnut', and 'Scatter'.

```
import React from 'react'
import { Line } from 'react-chartjs-2'
```

4. We can now write a graph component which creates the actual graph. Some points to note are that it will need to be defined within a single variable or function for later exporting. And, that it is returned with the graph specifying tag (<Line />).

```
const Demonstrate = () => {
  return (
    <div>
      <Line
        // Graph data
      />
    </div>
  )
}
```

Specific tag use descriptions and options can be found at:

<https://www.chartjs.org/docs/latest/samples/line/line.html>

5. Close the document with 'export default demonstrate' to allow access to the graph data from other files. Then returning to the main App.js file insert the same 2 lines from step 3 and add the further line:

```
import demonstrate from './newfolder/demonstrate'
```

This will import the given data from the previous file into the main App.

6. Insert the file tag (<Demonstrate />) at any point within the return's div element to display the graph

```
function App() {  
  return (  
    <div className="App">  
  
      <Demonstrate />  
  
    </div>  
  );  
}
```

Note: Throughout this process it is important to ensure the graph's .js file's name is capitalized on the first letter (Demonstrate as opposed to demonstrate) as this is how chart.js distinguishes special components.