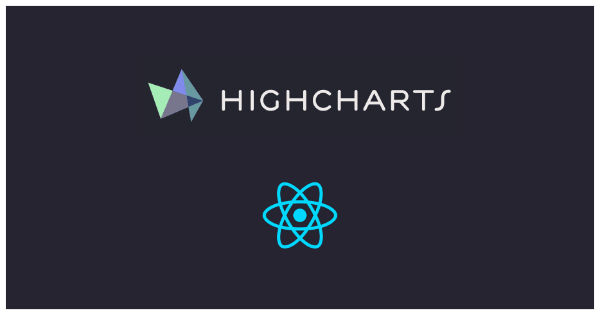
Frontend:



Highcharts là một thư viện biểu đồ được phát triển bằng JavaScript bởi Highsoft, một công ty được thành lập bởi Torstein Hønsi, cũng là người sáng tạo chính của Highcharts. Nó được phát hành lần đầu tiên vào năm 2009.

Highcharts có khả năng hiển thị các loại biểu đồ khác nhau, từ biểu đồ đường và cột cơ bản đến 3D và bản đồ nhiệt.

# 1. Cài đặt Highchart trong ReactJS

npm install react-highcharts --save-dev

Sau khi cài đặt hoàn tất, gói sẽ nằm trong thư mục node\_modules. Bây giờ chúng ta sẽ tạo một component để hiển thị biểu đồ , giả sử đặt tên file là Graph.js và import Highcharts ở đó.

import React from 'react';

import Highcharts from 'highcharts';

## **1.1. Vẽ đồ thị**

Ví dụ, chúng ta hãy chọn biểu đồ hình tròn và minh họa thành phần của bầu khí quyển Trái đất với: Nitơ (78,1%), Oxy (20,9%), Argon (0,9%) và Khí Trace (0,1%). Chúng ta sẽ render biểu đồ này trong div với id = atmospheric-composition.

render() {

return (

<div id="atmospheric-composition">

</div>

);

}

Bây giờ hay tạo một biến state là series và gán một mảng cho nó một array bao gồm dữ liệu của thành phần bầu khi quyển Trái đát được cấu trúc như sau:

constructor(props) {

super(props);

this.state = {

series: [{

name: 'Gases',

data: [

{

name: 'Argon',

y: 0.9,

color: '#3498db'

},

{

name: 'Nitrogen',

y: 78.1,

color: '#9b59b6'

},

{

name: 'Oxygen',

y: 20.9,

color: '#2ecc71'

},

{

name: 'Trace Gases',

y: 0.1,

color: '#f1c40f'

}

]

}]

}

}

Biểu đồ Highcharts được khởi tạo bằng cách sử dụng hàm tạo Highcharts.chart(). Toàn bộ đối tượng (trong dấu ngoặc nhọn) được truyền dưới dạng tham số cho nó được gọi là **Option** Object; và chart, title và plotOptions là các đối tượng trong khi đó serieslà một mảng. Để dễ dàng xử lý, hàm tạo Highcharts.chart() được đặt bên trong hàm highChartsRender() để chúng ta có thể gọi ra để khởi tạo hàm tạo. Bên trong đối tượng chart, giá trị pie tương ứng với thuộc tính type biểu thị rằng biểu đồ là biểu đồ hình tròn và giá trị chuỗi atmospheric-composition tương ứng với thuộc tính renderTo chính là id của phần tử bên trong biểu đồ sẽ được hiển thị. Và để làm cho biểu đồ hình tròn của chúng ta trông giống như một chiếc bánh rán, tùy chọn innerSize bên trong đối tượng pie của đối tượng plotOptions được đặt thành 70%. Đối tượng cuối cùng là series được gán biến this.state.series là một array.

highChartsRender() {

Highcharts.chart({

chart: {

type: 'pie',

renderTo: 'atmospheric-composition'

},

title: {

verticalAlign: 'middle',

floating: true,

text: 'Earth\'s Atmospheric Composition',

style: {

fontSize: '10px',

}

},

plotOptions: {

pie: {

dataLabels: {

format: '{point.name}: {point.percentage:.1f} %'

},

innerSize: '70%'

}

},

series: this.state.series

});

}

Vì chúng ta cần render biểu đồ ngay sau khi thành phần Graph.js được chèn vào DOM, nên chúng ta sẽ gọi hàm highChartsRender() bên trong componentDidMount().

componentDidMount() {

this.highChartsRender();

}

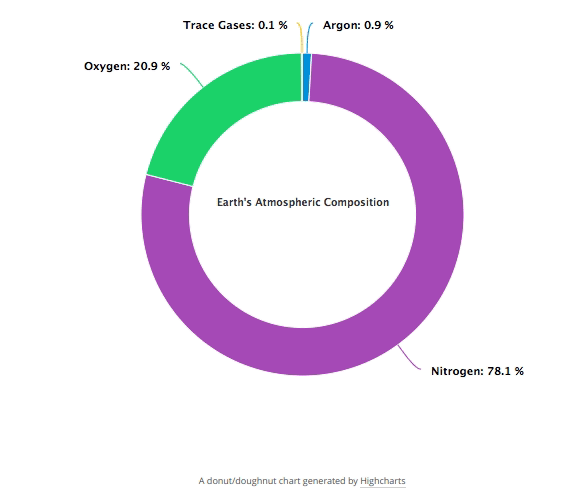
Bây giờ hãy import thành phần **Graph.js** bên trong file **index.js** của bạn

import Graph from './graph';

Và cuối cùng là render component Graph

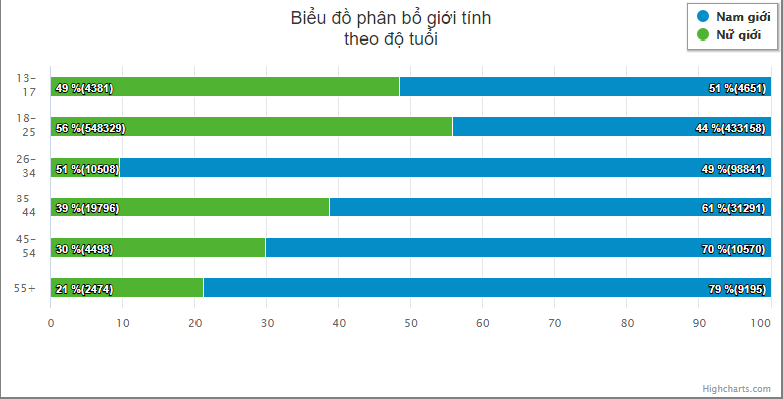
ReactDOM.render(<Graph/>, document.getElementById('root'));

Kết quả là:



1.2. Ứng dụng trong dự án

Điển hình: Biểu đồ phân bổ giới tính theo độ tuổi



# 2. Cách gọi API từ ReactJs

### 2.1 Cài đặt Axios để gọi API

Chạy lệnh:

npm install axios ---save

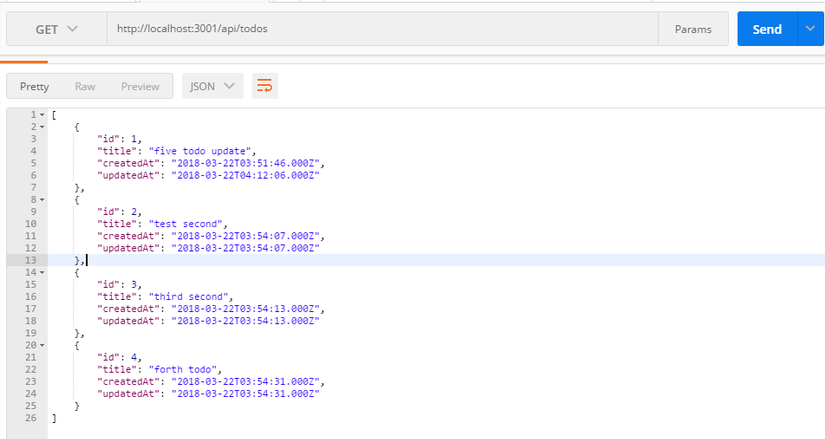
Code demo: [nodejs\_api](https://github.com/huuhv/nodejs_api)

Và fix port để chạy api, trong ví dụ này là 3001 chẳng hạn.

### 2.2. Danh sách items

#### Kết quả từ API trả về như sau:

URL: <http://localhost:3001/api/todos>



Trên frontend ta sẽ gọi bằng cách.

* File src/App.js mặc định sẽ là:

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

class App extends Component {

  render() {

    return (

      <div className="App">

        <header className="App-header">

          <img src={logo} className="App-logo" alt="logo" />

          <h1 className="App-title">Welcome to React</h1>

        </header>

        <p className="App-intro">

          To get started, edit <code>src/App.js</code> and save to reload.

        </p>

      </div>

    );

  }

}

export default App;

* Và file src/App.js, ta sẽ sửa file này bằng cách import thêm thư viện axios gọi phương thức api theo kiểu restful api. Thêm function componentDidMount để gọi api từ Nodejs API.

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

import axios from 'axios';

class App extends Component {

  constructor(){

    super();

    this.state ={items: []};

   }

  componentDidMount() {

    axios.get('http://localhost:3001/api/todos')

     .then(response => {

      console.log(response.data);

       this.setState({ items: response.data });

     })

     .catch(function (error) {

       console.log(error);

     })

    }

    render() {

      return (

          <div className="App">

              <h1>Todo List</h1>

              <div>

                <table className="table table-hover" align="center">

                  <thead>

                  <tr>

                      <th width="10%">ID</th>

                      <th width="80%">Title</th>

                      <th width="10%">Actions</th>

                  </tr>

                  </thead>

                  <tbody>

                    {this.state.items.map(item =>

                    <tr key={item.id}>

                      <td>{item.id}</td>

                      <td>{item.title}</td>

                      <td><a href="#">Edit</a>&nbsp;<a href="#">Delete</a></td>

                    </tr>

                    )}

                  </tbody>

                </table>

              </div>

          </div>

      );

  }

}

export default App;

Kết quả:



2.3. ứng dụng trong dự án

Gọi dữ liệu hiển thị trên frontend

3. Google map React

3.1

google-map-react là một thành phần được viết trên một bộ nhỏ **API Google Maps** . Nó cho phép bạn kết xuất bất kỳ thành phần React nào trên Google Map. Nó hoàn toàn đẳng cấu và có thể kết xuất trên máy chủ. Ngoài ra, nó có thể kết xuất các thành phần bản đồ trong trình duyệt ngay cả khi API Google Maps không được tải. Nó sử dụng thuật toán di chuột nội bộ, có thể điều chỉnh mọi đối tượng trên bản đồ đều có thể di chuột được.

Với 1 ví dụ đơn giản bạn chỉ cần thêm lat với lng props đến GoogleMapReact component

Ví dụ trong jsbin:

import React, { Component } from 'react';

import GoogleMapReact from 'google-map-react';

const AnyReactComponent = ({ text }) => <div>{text}</div>;

class SimpleMap extends Component {

  static defaultProps = {

    center: {

      lat: 59.95,

      lng: 30.33

    },

    zoom: 11

  };

  render() {

    return (

      <div style={{ height: '100vh', width: '100%' }}>

        <GoogleMapReact

          bootstrapURLKeys={{ key: /\* YOUR KEY HERE \*/ }}

          defaultCenter={this.props.center}

          defaultZoom={this.props.zoom}

        >

          <AnyReactComponent

            lat={59.955413}

            lng={30.337844}

            text="My Marker"

          />

        </GoogleMapReact>

      </div>

    );

  }

}

export default SimpleMap;

3.1. cài đặt Google-map-react

npm:

npm install --save google-map-react

yarn:

yarn add google-map-react

## 3.2. Đặc trưng

### 3.2.1. Hoạt động với các thành phần của bạn

Thay vì các điểm đánh dấu mặc định của Google Maps, bóng bay và các thành phần bản đồ khác, bạn có thể hiển thị các thành phần phản ứng hoạt hình thú vị của mình trên bản đồ.

### 3.2.2. Kết xuất đẳng cấu

Nó ám trên máy chủ. (Công cụ tìm kiếm chào mừng) (bạn có thể tắt javascript trong các công cụ phát triển trình duyệt và tải lại bất kỳ trang ví dụ nào để xem cách nó hoạt động)

### 3.2.3. Vị trí thành phần được tính độc lập với API Google Maps

Nó kết xuất các thành phần trên bản đồ trước (và thậm chí không có) API Google Maps được tải.

### 3.2.4. Tải xuống API Google Maps theo yêu cầu

Không cần phải đặt một <script src= thẻ ở đầu trang. API Google Maps tải khi sử dụng GoogleMapReact thành phần đầu tiên .

### 3.2.5. Sử dụng API Google Maps

Bạn có thể truy cập Google Maps map và maps các đối tượng bằng cách sử dụng onGoogleApiLoaded, trong trường hợp này bạn sẽ cần phải đặt yesIWantToUseGoogleMapApiInternalsthànhtrue

...

const handleApiLoaded = (map, maps) => {

  // use map and maps objects

};

...

<GoogleMapReact

  bootstrapURLKeys={{ key: /\* YOUR KEY HERE \*/ }}

  defaultCenter={this.props.center}

  defaultZoom={this.props.zoom}

  yesIWantToUseGoogleMapApiInternals

  onGoogleApiLoaded={({ map, maps }) => handleApiLoaded(map, maps)}

>

  <AnyReactComponent

    lat={59.955413}

    lng={30.337844}

    text="My Marker"

  />

</GoogleMapReact>

3.3. Ứng dụng trong dự án

