**Tìm hiểu React Hook useState**

Trong bài viết này chúng ta sẽ cùng nhau đi tìm hiểu về **React Hook useState**. Từ phiên bản 16.8 trở đi React cho ra mắt một tính năng mới đó là React Hooks, nó cho phép chúng ta làm việc với state, life cycle, và hàng loạt tính năng khác trong một functional component.

React Hooks bao gồm rất 10 hooks khác nhau, trong phạm vi bài viết này chúng ta sẽ tìm hiểu và xây dựng các ví dụ cụ thể bằng một hooks quan trọng và được sử dụng rất nhiều đó là **useState**.

**Mục lục**

* [1. useState trong React](https://freetuts.net/react-hook-usestate-2576.html#goto-h2-0)
* [2. Xây dựng ví dụ](https://freetuts.net/react-hook-usestate-2576.html#goto-h2-1)
  + [Ẩn/Hiện một component](https://freetuts.net/react-hook-usestate-2576.html#goto-h3-0)
  + [Lấy dữ liệu từ API và hiển thị](https://freetuts.net/react-hook-usestate-2576.html#goto-h3-1)

**1. useState trong React**

**useState** là một hook cho phép chúng ta quản lý các state trong một functional component, Bằng cách gọi React.useState bên trong một functional compoenent, bạn đã có thể làm việc với state một cách nhanh chóng.

Để sử dụng useState, trước tiên chúng ta cần import nó vào component.

*Bài viết này được đăng tại [free tuts .net]*

import React, { useState } from "react";

Tiếp theo, bạn chỉ cần sử dụng useState bằng cú pháp:

|  |  |
| --- | --- |
| 1 | const [nameState, funcUpdate] = React.useState(defaultState) |

Ở đây chúng ta có 3 đối số cần chú ý đến:

* nameState: đây là gía trị mặc định của state.
* funcUpdate: function dùng để cập nhật state. Giả sử mình muốn cập nhật giá trị của state mình chỉ cần gọi funcUpdate('giá trị mới của state').
* defaultState: giá trị mặc định của state khi được khởi tạo lần đầu.

Khi bạn sử dung state trong một class component, state được khởi tạo đó luôn luôn là một object, chúng ta chỉ có thể lưu trữ giá trị trong object đó.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | // Khởi tạo một state trong  // class component  // Giá trị luôn là một object  this.state = {    website: 'freetuts.net'  } |

Nhưng với hooks, chúng ta có thể lưu trữ bất cứ kiểu dữ liệu nào trong state như object, number, string,....Khi bạn thực hiện gọi *useState* đồng nghĩa với nó là state đã được khởi tạo thành công.

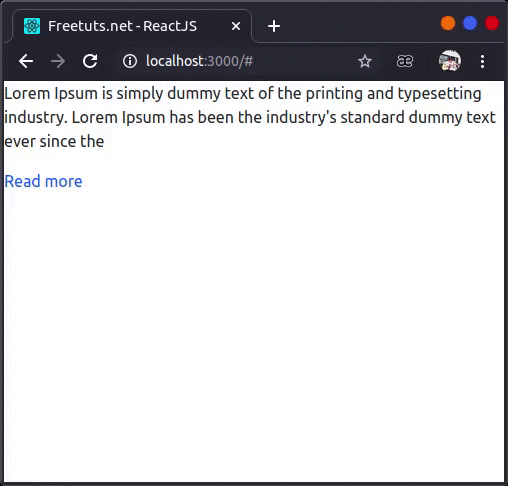
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | // Khởi tạo state trong functional component  // sử dụng React hook: useState    //Khởi tạo state có kiểu dữ liệu là string  const [website, setWebsite] = React.useState('freetuts.net');    //Kiểu dữ liệu là boolean  const [isLoading, setLoading] = React.useState(false);    ..... |

**2. Xây dựng ví dụ**

Chúng ta sẽ cùng nhau đi xây dựng ví dụ cụ thể, ở đây chúng ta sẽ từng bước đi xây dựng các ví dụ đơn giản nhất.

**Ẩn/Hiện một component**

một ví dụ cho phép ẩn/hiện các đoạn văn bản bằng cách click vào "đọc thêm".



Để triển khai ví dụ chúng ta sẽ thực hiện từng bước như sau:

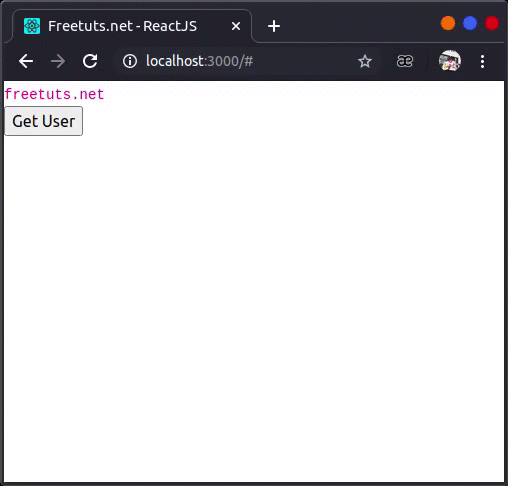
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | //Gọi React và useState  import React, { useState } from "react";    //Component hiển thi thêm nội dung  const MoreContent = () => {    return (      <p>          1500s, when an unknown printer took a galley of type and scrambled it to        make a type specimen book. It has survived not only five centuries, but        also the leap into electronic typesetting, remaining essentially        unchanged. It was popularised in the 1960s with the release of Letraset        sheets containing Lorem Ipsum passages, and more recently with desktop        publishing software like Aldus PageMaker including versions of Lorem        Ipsum.      </p>    );  };    export default function App(props) {    //Sử duụng useState    // isShow là một state    // setShow là function giúp cập nhật state    // Giá trị mặc định ban đầu của state là false    const [isShow, setShow] = useState(false);    return (      <div>        <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the</p>          {isShow === true ? <MoreContent /> : ''}        {isShow === false ? <a href="/#" onClick={() => {          // Khi click vào button          // cập nhật state bằng cách gọi hàm          // setShow đã được khai báo bên trên/          setShow(true)        }}>Read more</a> : ''}      </div>    );  } |

**Lấy dữ liệu từ API và hiển thị**

Để tiến hành triển khai ví dụ chúng ta cần tham khảo thêm các bài viết về:

* [Higher-Order Components trong ReactJS.](https://freetuts.net/higherorder-components-trong-reactjs-2434.html)
* [Tìm hiểu về Fragments trong ReactJS.](https://freetuts.net/fragments-trong-reactjs-2433.html)
* [Tìm hiểu về List và Keys trong ReactJS](https://freetuts.net/lists-va-keys-trong-reactjs-2423.html)

Sau khi đã hiểu hết về các phần kiến thức cần sử dụng, chúng ta sẽ tiến hành xây dựng ứng dụng cho phép lấy dữ liệu người dùng về từ API và hiển thị ra trình duyệt.



Trước tiên, chúng ta cần cài thêm một module cho phép gọi api có tên axios bằng cách sử dụng dòng lệnh:

|  |  |
| --- | --- |
| 1 | npm i axios |

Bạn có thể tham khảo thêm về module này bằng cách truy cập vào trang tài liệu chính thức của nó. Tiếp theo, chúng ta sẽ tạo một component và import các thư viện cần thiết vào:

|  |  |
| --- | --- |
| 1  2  3 | //Gọi React và useState  import React, { useState } from "react";  import axios from "axios"; //Sử dụng axios |

Khởi tạo component giúp hiển thị danh sách người dùng:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | //Component hiển thị danh sách người dùng  const ShowUser = (props) => {    //Lấy giá tri của props listUser    const { listUser } = props;      // Render ra list user    // React.Fragment cho phép bọc JSX lại.    // List Keys :  chỉ định key, giúp loại bỏ cảnh báo.    return (      <div>        {listUser.map((user, index) => {          return (            <React.Fragment key={user.id}>              <ul>                <li>{user.name}</li>                <li>{user.email}</li>              </ul>              <hr />            </React.Fragment>          );        })}      </div>    );  }; |

Sau đó khởi tạo component gọi API và hiển thị giao diện.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | export default function App(props) {    //Khai báo state, sử dụng hook: useState    const [listUser, setListUser] = useState([]);    const [isLoading, setLoading] = useState(false);      //API chứa dữ liệu người dùng    const getUserAPI =      "<https://5df8a4c6e9f79e0014b6a587.mockapi.io/freetuts/users>";      //Hàm fetch API để lấy dữ liệu ng. dùng    const getUser = () => {      //Cập nhật lại giá trị của state loading      setLoading(true);        //Thực hiện gọi api      axios        .get(getUserAPI)        .then((res) => {          //Cập nhật giá trị của state listUser          setListUser(res.data);        })        .catch((err) => {          //Trường hợp xảy ra lỗi          alert("Không thể kết nối tới server");        })        .finally(() => {          // Câu lệnh trong này được khởi chạy mỗi khi call API xong          // bất kể thất bại hay không.          // ...          setLoading(false); //Cập nhật giá trị của state isLoading        });    };      return (      <>  ​      <code>freetuts.net</code> <br />        {isLoading ? "loading..." : <button onClick={getUser}>Get User</button>}        <ShowUser listUser={listUser} />      </>    );  } |

Ở đây mình viết luôn vào file src/App.js của dự án nên sau khi gộp các phần ở trên, chúng ta sẽ có file App.js hoàn chỉnh:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70 | //Gọi React và useState  import React, { useState } from "react";  import axios from "axios"; //Sử dụng axios    //Component hiển thị danh sách người dùng  const ShowUser = (props) => {    //Lấy giá tri của props listUser    const { listUser } = props;      // Render ra list user    // React.Fragment cho phép bọc JSX lại.    // List Keys :  chỉ định key, giúp loại bỏ cảnh báo.    return (      <div>        {listUser.map((user, index) => {          return (            <React.Fragment key={user.id}>              <ul>                <li>{user.name}</li>                <li>{user.email}</li>              </ul>              <hr />            </React.Fragment>          );        })}      </div>    );  };    export default function App(props) {    //Khai báo state, sử dụng hook: useState    const [listUser, setListUser] = useState([]);    const [isLoading, setLoading] = useState(false);      //API chứa dữ liệu người dùng    const getUserAPI =      "<https://5df8a4c6e9f79e0014b6a587.mockapi.io/freetuts/users>";      //Hàm fetch API để lấy dữ liệu ng. dùng    const getUser = () => {      //Cập nhật lại giá trị của state loading      setLoading(true);        //Thực hiện gọi api      axios        .get(getUserAPI)        .then((res) => {          //Cập nhật giá trị của state listUser          setListUser(res.data);        })        .catch((err) => {          //Trường hợp xảy ra lỗi          alert("Không thể kết nối tới server");        })        .finally(() => {          // Câu lệnh trong này được khởi chạy mỗi khi call API xong          // bất kể thất bại hay không.          // ...          setLoading(false); //Cập nhật giá trị của state isLoading        });    };      return (      <>        <code>freetuts.net</code> <br />        {isLoading ? "loading..." : <button onClick={getUser}>Get User</button>}        <ShowUser listUser={listUser} />      </>    );  } |

Khởi chạy dự án bằng cách gõ dòng lệnh:

|  |  |
| --- | --- |
| 1 | npm start |

Trên đây chúng ta đã cùng nhau đi tìm hiểu về useState trong ReactJS Hook. Đây là kiến thức rất cơ bản về nó nhưng cũng hết sức quan trọng trong quá trình làm việc với ReactJS sau này. Mong rằng bài viết sẽ giúp ích cho bạn.