**Tìm hiểu State trong ReactJS**

Trong bài viết này chúng ta sẽ cùng nhau đi **tìm hiểu về khái niệm state trong ReatJS**. ReactJS được xây dựng để làm các trang web *single page applicationn (SPA)* nên vấn re-render lại các component là điều tất yếu và không thể thiếu. **State** cho phép chúng ta lưu trữ dữ liệu trong component, và mỗi khi **state** thay đổi thì component đó sẽ tự động re-render.

**Mục lục**

* [State trong ReactJS là gì ?](https://freetuts.net/state-trong-reactjs-2358.html#goto-h2-0)
* [Thao tác với state trong ReactJS](https://freetuts.net/state-trong-reactjs-2358.html#goto-h2-1)
  + [Khởi tạo môt state](https://freetuts.net/state-trong-reactjs-2358.html#goto-h3-0)
  + [Cập nhật một state](https://freetuts.net/state-trong-reactjs-2358.html#goto-h3-1)
* [Sự khác nhau giữa props và state](https://freetuts.net/state-trong-reactjs-2358.html#goto-h2-2)

**State trong ReactJS là gì ?**

**State** là một object có thể được sử dụng để chứa dữ liệu hoặc thông tin về components. State có thể được thay đổi bất cứ khi nào mong muốn. Khác với props bạn có thể truyền props sang các components khác nhau thì state chỉ tồn tại trong phạm vi của components chứa nó, mỗi khi state thay đổi thì components đó sẽ được render lại.

Trong các dự án React, state được dùng để phản hồi các yêu cầu từ người dùng, hay lưu trữ một dữ liệu nào đó trong components.

**Thao tác với state trong ReactJS**

Chúng ta có thể thao tác với **state** trong một component rất dễ dàng bằng cách sử dụng class components. Bên dưới mình sẽ chỉ ra các thao tác với state trong một component.

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

**Khởi tạo môt state**

Chúng ta có thể khởi tạo một state bằng cách gán giá trị cho biến this.state:

|  |  |
| --- | --- |
| 1 | this.state = { name : 'freetuts.net' } |

và lấy gía trị của state bằng cách gọi this.state:

|  |  |
| --- | --- |
| 1 | console.log(this.state.website) //freetuts.net |

Ví dụ bên dưới mình có một class components và mình sẽ tiến hành khởi tạo state bên trong phương thức constructor :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | import React from "react";    class App extends React.Component {    constructor(props) {      super(props);      //Chỉ định một state      this.state = { website: "Freetuts.net" };    }    render() {      return (        <div>          <h1>Học ReactJS căn bản tại {this.state.website} </h1>        </div>      );    }  }  export default App; |

Trong hầu hết các trường hợp bạn nên khởi tạo state bên trong hàm constructor() để tránh gặp các lỗ không mong muốn. Vì đây sẽ là hàm khởi chạy đầu tiên khi một components được gọi.

**Cập nhật một state**

Trong các components bạn cần phải thao tác với state rất nhiều , ngoài thêm và lấy giá trị của state bạn còn phải cập nhật các states để ReactJS có thể tự động re-render lại components. Điều này khá quan trọng, giả sử bạn đang cho người dùng nhập vào môt form nào đó và khi click *Lưu* thì nội dung được điền trong form lúc này sẽ phải hiển thị ra màn hình. Đây là lúc bạn cần dùng đến *cách thay đổi giá trị của một state*.

Để **cập nhật một state** bạn sử dụng phương thức:

|  |  |
| --- | --- |
| 1  2  3 | this.setState({       name : 'newValue'  }) |

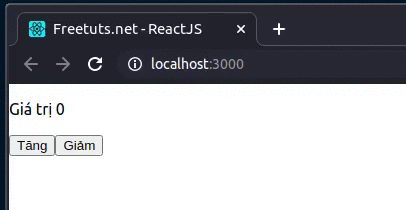
Ngoài ra bạn cũng có thể lấy giá trị của state trước khi cập nhật:

|  |  |
| --- | --- |
| 1  2  3 | this.setState((state) => {        return newValue;  }); |

Bên dưới mình có một ví dụ về cập nhật state index khi nhấn vào click vào button tương ứng :

|  |  |
| --- | --- |
| 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 | import React from "react";    class App extends React.Component {    constructor(props) {      super(props);      //Chỉ định một state      this.state = { index: 1 };    }    render() {      return (        <div>          <p>Giá trị {this.state.index}</p>          <button            onClick={() => {              this.setState({                index: this.state.index + 1              })            }}          >            Tăng          </button>          <button            onClick={() => {              this.setState({                index: this.state.index -1              })            }}          >            Giảm          </button>        </div>      );    }  }  export default App; |

Chúng ta sẽ thấy giá trị của state sẽ được thay đổi mỗi khi bạn click vào buttonTăng hoặc Giảm :



**State** là một khái niệm khá đơn giản nhưng cũng hết sức quan trọng trong component, mặc dù props và state hay đi đôi với nhau nhưng nó hoàn toàn khác biệt với nhau.

**Sự khác nhau giữa props và state**

Trong quá trình học React, cũng có vài trường hợp bạn chưa hiểu về 2 khái niệm props và state kĩ nên có sự nhầm lẫn ở đây.

* **State -**Dữ liệu chỉ nằm trong phạm vi của một component. Nó được sở hữu bởi một components cụ thể mà chỉ là của component đó thôi. Ví dụ, như người yêu bạn chỉ là của bạn vậy =))). Và mỗi khi state thay đổi thì component cũng phải thay đổi theo.
* **Props -**Dữ liệu đường truyền từ component cha cho componet con, components con khi nhận được sẽ chỉ được đọc mà không thể thay đổi dữ liệu đó.

Sự khác nhau chính của 2 khái niệm này là component sở hữu dữ liệu. State là chỉ riêng nó có thể sử dụng. Props là dữ liệu mà component con được nhận về từ một component cha.

Vì phạm vi của state chỉ nằm trong components nên việc truyền dữ liệu từ các components với nhau người ta thường dùng props. Nhưng vấn đề ở đây là props chỉ có thể truyền cho componet con của nó và khi truyền cho các componet cháu, chắt khá rắc rối. Bởi vậy chúng ta có thêm khái niệm về Redux, mình sẽ giới thiệu ở phần tiếp theo vì đây là phần khá phức tạp và đau não :))

Trong bài này chúng ta đã cùng nhau đi tìm hiểu về state trong ReactJS. 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.