**Tìm hiểu về Render Props trong ReactJS**

Trong bài này chúng ta sẽ cùng nhau đi **tìm hiểu về Render Props trong ReactJS**. Khi triển khai một dự án React, việc tái sử dụng các component là điều rất cần thiết nhất là trong các dự án lớn. Bởi vậy, React cung cấp cho chúng ta một pattern rất hữu ích cho việc xây dựng và tái sử dụng các component đó là render props.

**Mục lục**

* [1. Render props trong ReactJS](https://freetuts.net/render-props-trong-reactjs-2432.html#goto-h2-0)
* [2. Triển khai ví dụ về render props trong React](https://freetuts.net/render-props-trong-reactjs-2432.html#goto-h2-1)
  + [Ví dụ 1](https://freetuts.net/render-props-trong-reactjs-2432.html#goto-h3-0)
  + [Ví dụ 2](https://freetuts.net/render-props-trong-reactjs-2432.html#goto-h3-1)

**1. Render props trong ReactJS**

**Render props** trong React là một kỹ thuật để tái sử dụng các đoạn mã, có mục đích tương tự với phương pháp sử dụng Higher Order Component, giúp chúng ta sử dụng lại logic trên nhiều component.

Nó thực hiện bằng cách *truyền vào component một props có value như là một function*. **Render props** được sử dụng trong rất nhiều các module nổi tiếng trong hệ sinh thái của react, bao gồm cả *react-router*. Chúng ta sẽ đi vào cách triển khai sau đó làm một vài ví dụ nhỏ để hiểu rõ hơn.

Để sử dụng kỹ thuật render props. Trước tiên, chúng ta sẽ truyền vào component 1 props là một function, và lúc này chúng ta sẽ nhận được giá trị trả về.

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

|  |  |
| --- | --- |
| 1 | return <Freetuts render={(data) => (<p>welcome to {data.value}</p>)} />; |

và trong component được gọi (ví dụ trên là component Freetuts), sẽ gọi props render và trả về giá trị cần render và giá trị cần trả về vào hàm props.render().

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | const Freetuts = (props) => {    return (      <div>        {props.render({          value: "freetuts.net"        })}      </div>    );  }; |

Lúc này chúng ta đã triển khai thành công kỹ thuật render props và đây là kết quả :

|  |  |
| --- | --- |
| 1  2  3 | <div>        <p>welcome to freetuts.net</p>  </div> |

Chắc đến đây các bạn cũng chưa hiểu rõ cách hoạt động của nó. Chúng ta sẽ đi vào ví dụ cụ thể bên dưới.

**2. Triển khai ví dụ về render props trong React**

Ở đây mình sẽ đi xây dựng 2 ví dụ cụ thể đề các bạn hiểu rõ hơn về render props.

**Ví dụ 1**

Chúng ta sẽ dùng xây dựng một component nhận về một object chứa danh sách người yêu của bạn sau đó hiển thị ra :)) Ở đây mình sẽ sử dụng render props để có thể tái sử dụng đoạn code nhiều lần.

Trước tiên, chúng ta sẽ đi xây dựng component có tên ShowGirlFriends

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | const ShowGirlFriends = (props) => {    return (      <ul>        {props.listGirlFriends.map((person, index) => {          props.children(person);          return <li key={person.id}>{person.name}</li>;        })}      </ul>    );  }; |

component này có nhiệm vụ lấy giá trị của props listGirlFriends su đó sẽ dùng map để render lần lượt từng người, và trả về cho props children giá trị của người hiện tại.

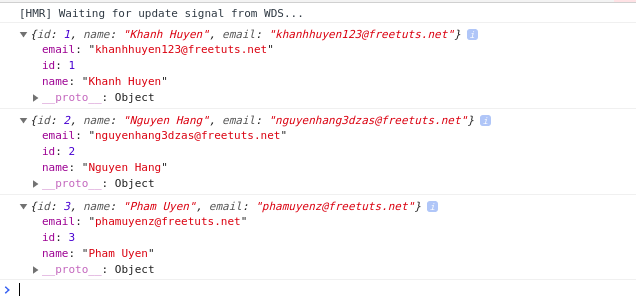
Sau đó, chúng ta chỉ cần gọi component ShowGirlFriends ở bất cứ đâu bạn muốn nó hiển thị.

|  |  |
| --- | --- |
| 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 | function App(props) {    //Object chứa thông tin của người iwww :))    const myGirlFriends = [      {        id: 1,        name: "Khanh Huyen",        email: "khanhhuyen123@freetuts.net",      },      {        id: 2,        name: "Nguyen Hang",        email: "nguyenhang3dzas@freetuts.net",      },      {        id: 3,        name: "Pham Uyen",        email: "phamuyenz@freetuts.net",      },    ];      return (      <ShowGirlFriends listGirlFriends={myGirlFriends}>        {(data) => {          //Nhận data từ component ShowGirlFriends khi nó trả về          //bằng đoạn props.children(person)          console.log(data);        }}      </ShowGirlFriends>    );  } |

Lúc này chúng ta sẽ thấy kết quả, các đoạn mã html khi được render sẽ như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <ul>    <li>Khanh Huyen</li>    <li>Nguyen Hang</li>    <li>Pham Uyen</li>  </ul> |

và đồng thời bạn sẽ nhận được giá trị của mỗi người được trả về thông qua props children:



**Ví dụ 2**

Ở ví dụ tiếp theo này chúng ta sẽ đi xây dựng bộ đếm, bao gồm nút tăng giảm..

Trước tiên, chúng ta sẽ đi xây dựng một component có tên Counter, có nhiệm vụ xây dựng hàm tăng, giảm và hiển thị số.

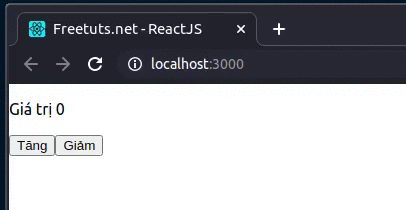
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | class Counter extends React.Component {    constructor(props) {      super(props);      this.state = {        count: 0,      };      //Bind this      this.increment = this.increment.bind(this)      this.decrement = this.decrement.bind(this)    }    //Hàm này sẽ TĂNGiá trị của số    increment() {      this.setState({        count: this.state.count + 1,      });    }    //Hàm này sẽ GIẢM giá trị của số    decrement() {      this.setState({        count: this.state.count - 1,      });    }    render() {    return <div>{      //Trả về giá trị cho props render.      this.props.render({        count: this.state.count,        increment: this.increment,        decrement: this.decrement      })    }</div>;    }  } |

Ở đây chúng ta sẽ trả về giá trị của số, và hàm thực hiện tăng giảm số vào props có tên render.

Tiếp theo, mình sẽ gọi component này ở bất cứ đâu, props có tên render được tryền vào sẽ có giá trị là một function và tham số của function đó sẽ là giá trị trả về.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | default function App(props) {    return (      <Counter        render={(data) => {          //Nhận giá trị trả vê từ Counter          //qua props render.          const { count, increment, decrement } = data;          return (            <>              <p>Giá trị {count}</p>              <button onClick={increment}>Tăng</button>              <button onClick={decrement}>Giảm</button>            </>          );        }}      />    );  } |

Lúc này mình sẽ hiển thị count và các nút tăng giảm, khi bạn click vào nút tăng giảm nó sẽ gọi hàm tương ứng. Và đây là kết quả :



Trên đây chúng ta đã cùng nhau đi tìm hiểu về Render Props 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.