**Tìm hiểu về Refs trong ReactJS**

Trong bài viết này chúng ta sẽ cùng nhau đi **tìm hiểu về refs trong ReactJS**. Trong quá trình làm việc với React ở một vài trường hợp nào đó bạn cần phải thực hiện việc tham chiếu đến DOM, mặc dù điều này không được khuyến khích sử dụng. Để giúp quá trình tham chiếu đến DOM dễ dàng hơn, React cung cấp cho chúng ta **React Refs.**

**Mục lục**

* [1. React Refs là gì?](https://freetuts.net/refs-trong-reactjs-2430.html#goto-h2-0)
* [2. Sử dụng React Refs](https://freetuts.net/refs-trong-reactjs-2430.html#goto-h2-1)
  + [Khởi tạo một Ref](https://freetuts.net/refs-trong-reactjs-2430.html#goto-h3-0)
  + [Forwarding Refs](https://freetuts.net/refs-trong-reactjs-2430.html#goto-h3-1)

**1. React Refs là gì?**

**React refs** là một tính năng hữu ích, nó là cách mà chúng ta có thể để tham chiếu một element trong DOM hoặc từ một class component con đến component cha. Điều này cho phép chúng ta đọc và chỉnh sửa các element đó.

Cách giải thích dễ hiểu nhất về React Refs là tưởng tượng nó là một cây cầu. Khi một element được gán vào một ref nó sẽ cho phép chúng ta sửa đổi cũng như truy cập vào element đó ngay lập tức và không cần sử dụng đến props hay state để component re-render lại. Nó giống cho phép việc can thiệp vào DOM như trong Javascript DOM:

document.getElementsByTagName('h1').innerHTML = 'Freetuts.net'

Chúng ta có thể can thiệp trực tiếp vào DOM qua refs mà không cần thông qua việc render. Mặc dù đây là cách để can thiệp vào DOM thuận tiện mà không cần phải sử dụng đến state, props nhưng điều này React không khuyến khích. Bởi khi các DOM bị thay đổi thì nó sẽ ảnh hưởng một phần nào đó đến quá trình render các component. Các bạn nên sử dụng **React refs** để can thiệp vào DOM trong trường hợp cần thiết.

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

**2. Sử dụng React Refs**

Trong phần này chúng ta sẽ cùng nhau đi tìm hiểu về cách làm việc với React Refs trong React

**Khởi tạo một Ref**

**Refs** thường được chỉ định trong hàm tạo trong constructor ở class component và như một biến ở functional component. Sau đó được gắn vào một element trong hàm render(). Ở đây chúng ta sẽ tạo một **refs** và gắn vào element input :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | import React from "react";      export default class MyComponent extends React.Component {    constructor(props) {      super(props);      //Khởi tạo một ref      this.myRef = React.createRef();    }      render() {      return (        <input          name="email"          onChange={this.onChange}          ref={this.myRef}          type="text"        />      );    }  } |

Ở ví dụ bên trên mình đã khởi tạo một ref bằng cách sử dụng hàm React.createRef() và gán giá trị của nó vào thuộc tính myRef trong class MyComponent. Lúc này, chúng ta chỉ cần gán refs vào element input. Khi một **ref** được gắn vào một element, element đó có thể được truy cập và sửa đổi thông qua ref. Cụ thể trong trường hợp này là input

Tiếp theo, mình sẽ thêm một **button** cho ví dụ bên trên, khi click vào button đó nó sẽ tham chiếu đến input qua refs và thực hiện focus input đó.

...

handleClick = () => {

//thuộc tính current trong refs cho phép

//chúng ta chỉ định element hiện tại được tham chiếu.

this.myRef.current.focus();

}

render() {

return (

...

<button onClick={this.handleClick}>

Focus Input

</button>

</>

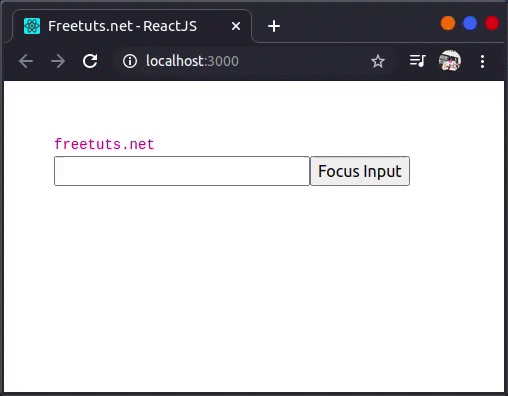
)

}

Thuộc tính current trong myRefs chứa giá trị element được tham chiếu khi element đó được render. Bằng cách sử dụng React ref chúng ta sẽ thay đổi trạng thái của input mà không cần phải sử dụng đến state hay props. Trong trường hợp, bạn chỉ muốn focus một input hay làm mờ hình ảnh thì bạn chỉ cần sử dụng refs mà không cần phải re-render lại component để chỉ làm những việc này.

Cuối cùng, chúng ta sẽ có ví dụ đầy đủ:

|  |  |
| --- | --- |
| 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 | import React from "react";      export default class MyComponent extends React.Component {    constructor(props) {      super(props);      this.myRef = React.createRef();    }    handleClick = () => {      this.myRef.current.focus();    }    render() {      return (        <>        <code>freetuts.net</code>        <input          name="email"          onChange={this.onChange}          ref={this.myRef}          type="text"        />        <button onClick={this.handleClick}>          Focus Input        </button>        </>      );    }  } |



**Forwarding Refs**

**Forwarding Refs** là một kỹ thuật để tự động chuyển một ref từ một component tới component con, cho phép component cha có thể tham chiếu tới các element của component con để đọc và chỉnh sửa nó.

React cung cấp cho chúng ta một cách thức để thực hiện việc chuyển tiếp một ref, chúng ta sẽ bao component con trong React.forwardRef(), ở đây mình có ví dụ:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | //Component Con  const MyInput = React.forwardRef((props, ref) => {     return(<input name={props.name} ref={ref} />);  });  // Component Cha  const MyComponent = () => {     let ref = React.createRef();     return (       <MyInput         name="email"         ref={ref}       />     );  } |

Ví dụ bên trên mình đã sử dụng React.forwardRef(), ở đây nó cung cấp cho chúng ta 2 tham số lần lượt là props và refs, cho chúng ta nhận về giá trị của **props** và **refs** từ component cha.

Chúng ta có thể sử dụng React.forwardRef() trong class component bằng cách này sử dụng HOC (higher order component)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | // Lấy ref thông qua props  class WrappedComponent extends Component {    render() {      return (        <input          type="text"          name={this.props.name}          ref={this.props.innerRef}         />      )    }  }  // Gói componentnt được bọc của chúng tôi với ForwardRef, truyền vào props giá trị của refs  const MyInput = React.forwardRef((props, ref) => {    return (<WrappedComponent innerRef={ref} {...props} />);  });  export default MyInput; |

Và cuối cùng chúng ta sẽ làm ví dụ có chức năng giống ví dụ ban đầu nhưng chúng ta sẽ tham chiếu đến element input trong component con.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | import React from "react";    //Component Con  const MyInput = React.forwardRef((props, ref) => {    return <input name={props.name} ref={ref} />;  });  // Component Cha  const MyComponent = () => {    let ref = React.createRef();    const handleButton = () => {      ref.current.focus();    };    return (      <>        <code>freetuts.net</code>        <MyInput name="email" ref={ref} />        <button onClick={handleButton}>Focus Input</button>      </>    );  };  export default MyComponent; |

Kết quả cũng tương tự như ví dụ ban đầu.

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