**Handling Events (xử lý sự kiện) trong ReactJS**

Trong bài viết này chúng ta sẽ cùng nhau đi **tìm hiểu về Handling Events trong Reactjs** - xử lý sự kiện trong ReactJS. Trong một website việc tương tác giữa người dùng là điều không thể thiếu như click, nhập form,..chúng ta có thể thực hiện bắt các sự kiện này trong React một cách dễ dàng.

**Mục lục**

* [1. Handling Events](https://freetuts.net/handling-events-trong-reactjs-2393.html#goto-h2-0)
* [2. Lưu ý với this trong xử lý Events](https://freetuts.net/handling-events-trong-reactjs-2393.html#goto-h2-1)
  + [Xây dựng ví dụ](https://freetuts.net/handling-events-trong-reactjs-2393.html#goto-h3-0)

**1. Handling Events**

Xử lý các sự kiện trong React rất giống với xử lý các sự kiện trên các phần tử DOM. Có một số khác biệt về cú pháp:

* Các sự kiện React được đặt tên bằng camelCase, thay vì chữ thường. Ví dụ: *onclick -> onClick, onchange -> onChange*
* Với JSX, bạn truyền một hàm để bắt sự kiện, thay vì một chuỗi như HTML thông thường.

Ở đây mình có ví dụ với HTML

|  |  |
| --- | --- |
| 1  2  3 | <button onclick="changeName()">    Change Name  </button> |

khi làm việc với JSX trong React chúng ta sẽ phải viết như sau :

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

|  |  |
| --- | --- |
| 1  2  3 | <button onClick={changeName}>    Change Name  </button> |

Một điểm khác biệt nữa là bạn không thể sử dụng return false để chặn các hành động mặc định được (prevent default), trong React bạn cần phải sử dụng preventDefault(). Giả sử trong HTML mình muốn dừng hành động mặc định của một form:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <form onsubmit="console.log('form submit'); return false;">      <div className="form-group">          <label htmlFor="text">Email:</label>          <input type="text" className="form-control" name="email" placeholder="Enter email" onChange={e=>          this.changeInputValue(e)}          />      </div>      <button type="submit" className="btn btn-primary">          Submit      </button>      </div>  </form> |

trong React, chúng ta cần phải sử dụng e.preventDefault():

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | function ActionLink() {      function submitForm(e) {        e.preventDefault();        console.log('form submit !!');      }        return (       <form onSubmit = {(e) => submitForm(e)}>           <button type="submit">Submit</button>       </form>      );    } |

Ở đây, e là một object chứa tất cá event. React định nghĩa object này theo [W3C spec](https://freetuts.net/handling-events-trong-reactjs-2393.html), vì vậy chúng ta không cần phải quan tâm về khả năng tương thích giữa các trình duyệt với nhau.

**2. Lưu ý với this trong xử lý Events**

Ở đây mình có một ví dụ, khi click vào button tương ứng sẽ thực hiện ẩn/hiển nội dung:

|  |  |
| --- | --- |
| 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 | import React from "react";    class App extends React.Component {    constructor(props) {      super(props);      this.state = {        isShow: true      };    }    toggleMSG() {      this.setState({        isShow: !this.state.isShow      });    }    render() {      return (        <div>         <b>Nội dung : {this.state.isShow ? "Freetuts.net - Lập trình ReactJS" : ""}</b>  <br />          <button onClick={() => this.toggleMSG()}>            {this.state.isShow ? "HIDE" : "SHOW"}          </button>        </div>      );    }  }  export default App; |

Trong sự kiện onClick chúng ta thực hiện gọi hàm toggleMSG(), hãy cẩn thận trong khi gọi hàm này, bạn cần phải truyền vào đó biến this, bằng các cách viết như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | //Sử dụng arrow function  <button onClick={() => this.toggleMSG()}>    //Sử dụng bind  <button onClick={this.toggleMSG.bind(this)}>    //hoặc bind this trong constructor() |

**Xây dựng ví dụ**

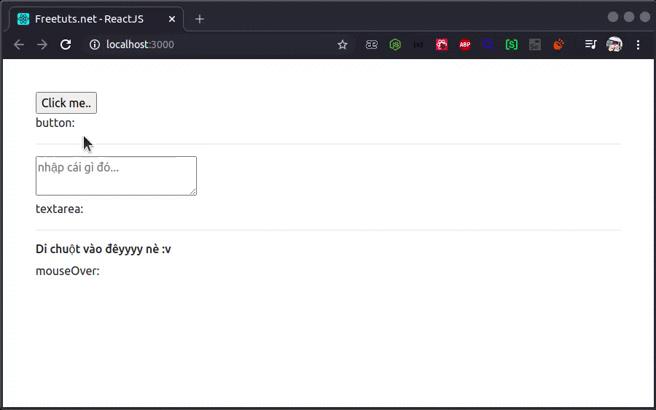
Trong phần này mình sẽ đi xây dựng ví dụ về bắt sự kiện trong React và một vài cách gọi hàm trong React. Trong file src/App.js chúng ta có ví dụ:

|  |  |
| --- | --- |
| 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 | import React from "react";    class App extends React.Component {    constructor(props) {      super(props);      //Khởi tạo inital state      this.state = {        textareaChange: "",        buttonClick : "",        mouseOver: ""      };      //bind this cho function mouseOver để  tránh gặp lỗi      //không tồn tại biến this.      this.mouseOver = this.mouseOver.bind(this)    }    changeText(e) {      this.setState({        textareaChange: e.target.value      });    }    mouseOver() {      this.setState({        mouseOver: this.state.mouseOver + "mouseOver..."      });    }    render() {      return (        <div style={{marginTop: "5%"}}>          <button            onClick={() => {              this.setState({                buttonClick: this.state.buttonClick + "onClick..."              });            }}          >            Click me..          </button>{" "}            <p>            button: <code>{this.state.buttonClick}</code>          </p>          <hr />          <textarea onChange={e => this.changeText(e)} placeholder="nhập cái gì đó..."></textarea>          <p>            textarea: <code>{this.state.textareaChange}</code>          </p>            <hr />          <h6 onMouseOver={this.mouseOver}>Di chuột vào đêyyyy nè :v</h6>          <p>            mouseOver: <code>{this.state.mouseOver}</code>          </p>        </div>      );    }  }    export default App; |

Bên trên mình xây dựng 2 hàm changeText() và mouseOver(), khi sự kiện được gọi nó sẽ gọi hàm tương ứng. Trong sự kiện onClick mình thực hiện viết câu lệnh trực tiếp mà không cần gọi hàm. Chúng ta khởi chạy ví dụ bằng cách sử dụng dòng lệnh:

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

và đây là kết quả của chúng ta:



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