

Bài giảng

PHÁT TRIỂN ỨNG DỤNG WEB

Lê Đình Thanh

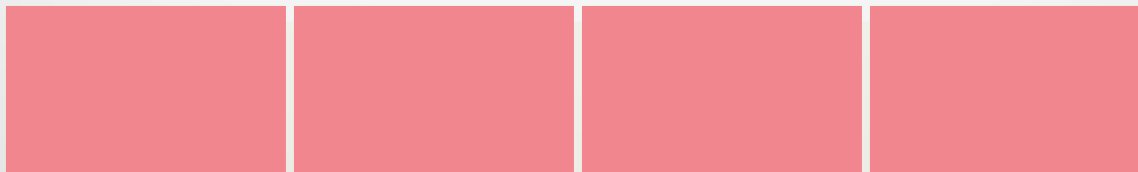
Khoa Công nghệ Thông tin

Trường Đại học Công nghệ, ĐHQGHN

E-mail: thanhld@vnu.edu.vn Mobile: 0987.257.504

Chương 5

Thư viện phát triển mặt trước



Nội dung

- jQuery
- Bootstrap
- **React**

Giới thiệu React

- Thư viện phát triển giao diện người dùng được Facebook phát triển
 - <https://facebook.github.io/react>
- Phân tách giao diện thành các thành phần (component) độc lập, có thể sử dụng lại.
 - Chấp nhận dữ liệu vào bất kỳ và trả về các phần tử React miêu tả giao diện.
 - Thư viện React sẽ chuyển đổi các phần tử React thành các đối tượng DOM.

Thành phần/lớp React

- Thành phần React được định nghĩa bằng lớp Javascript kế thừa *React.Component*
 - Phương thức *render()*
 - trả về các phần tử React, hay các phần tử giao diện.
 - Thuộc tính *props*
 - là đối tượng Javascript đơn giản (không có phương thức)
 - chứa dữ liệu vào cho thành phần
 - chỉ đọc, được xác định tại thời điểm thành phần được tạo
 - Thuộc tính *state*
 - là đối tượng Javascript đơn giản
 - nắm giữ những dữ liệu bên trong, riêng của thành phần
 - mỗi khi giá trị của *state* thay đổi, thành phần sẽ cập nhật lại giao diện, tức gọi phương thức *render()*
 - Phương thức *setState()*
 - Được dùng để thay đổi giá trị của *state*

React.Component
(*abstract base class*)



AbcGUIComponent

+ render()
+ setState()
- props
- state

Phần tử React

- Mỗi phần tử React được chuyển đổi thành một đối tượng DOM
 - React virtual DOM được chuyển thành HTML DOM
- Phương thức `React.createElement(type, [props], [children])`
 - được sử dụng để khai báo các phần tử React
 - `type` có thể là tên thẻ (như HTML) hoặc tên lớp React

Đưa thành phần React lên giao diện

- Phương thức `ReactDOM.render(reactComp, domNode)`
 - `reactComp` là thành phần React
 - `domNode` là đối tượng DOM
 - `reactComp` được chuyển đổi thành nhánh các đối tượng DOM là con, cháu của đối tượng `domNode`.

Ví dụ đơn giản

```
<style> .box {border:solid 1px red;} </style>
<div id="root"></div>
<script>
    class Product extends React.Component {
        constructor(props) { super(props); }
        render() {
            return React.createElement("div", {className:"box"},
                React.createElement("h1", null, this.props.name),
                React.createElement("p", null, this.props.description)
            );
        }
    }
    ReactDOM.render(
        React.createElement(Product,
            {name:"Dell Laptops", description:"Laptops from Dell"}
        ),
        document.getElementById('root')
    );
</script>
```

Dell Laptops

Laptops from Dell

Cập nhật giao diện, xử lý sự kiện

- Thay đổi *state* dẫn đến cập nhật giao diện
 - Gọi `setSate()`
- Xử lý sự kiện
 - Gán hàm Javascript cho con trỏ sự kiện của phần tử React

Cập nhật giao diện, xử lý sự kiện

```
class Product extends React.Component {
  constructor(props) {
    super(props);
    this.state = {isBookmarked:false}
    this.handleClick = this.handleClick.bind(this);
  }
  handleClick() {
    this.setState(prevState =>
      ({ isBookmarked: !prevState.isBookmarked }));
  }
  render() {
    return React.createElement( "div",
      {className: this.state.isBookmarked ? 'box yellow' : 'box'},
      React.createElement("h1", null, this.props.name),
      React.createElement("p", null, this.props.description),
      React.createElement("button", { onClick: this.handleClick },
        this.state.isBookmarked ?
          'Remove bookmark' : 'Set bookmark')
    );
  }
}
```

Dell Laptops

Laptops from Dell

Set bookmark

Dell Laptops

Laptops from Dell

Remove bookmark

Buộc dữ liệu một-chiều, trên-xuống

- Thành phần có thể chứa thành phần con
- Luồng/buộc dữ liệu một-chiều, trên-xuống
 - Dữ liệu từ thành phần cha được chuyển xuống thành phần con thông qua *props*, rồi từ thành phần con xuống thành phần cháu, ...

Buộc dữ liệu một-chiều, trên-xuống

```
ReactDOM.render(  
  React.createElement(ProductList, { products:  
    [{ name: "Dell Laptops", description: "Laptops from Dell", isBookmarked: false },  
      { name: "Dell PCs", description: "PCs from Dell", isBookmarked: true },  
      { name: "HP Laptops", description: "Laptops from HP", isBookmarked: false }] }  
    ),  
  document.getElementById('root')  
);  
class ProductList extends React.Component {  
  constructor(props) { super(props); }  
  render() {  
    const listItems = this.props.products.map(p =>  
      React.createElement(Product,  
        { name: p.name, description: p.description,  
          isBookmarked: p.isBookmarked }));  
    return React.createElement('div', null, listItems);  
  }  
}  
class Product extends React.Component {  
  ...  
  React.createElement("h1", null, this.props.name),  
  ...  
}
```



The diagram illustrates the data flow in a one-way binding. A red arrow originates from the `products` prop in the `ProductList` component's props object. It points to the `this.props.products` access within the `render` method of `ProductList`. From there, another red arrow points to the `name` prop being passed to the `Product` component's `createElement` call. Finally, a third red arrow points from the `this.props.name` access in the `Product` component to the `name` prop of the `createElement` call for the `h1` tag in the `Product` component's render method. This visualizes how data is passed from parent to child components.

Dell Laptops

Laptops from Dell

Set bookmark

Dell PCs

PCs from Dell

Remove bookmark

HP Laptops

Laptops from HP

Set bookmark

Buộc dữ liệu một-chiều, trên-xuống

```
ReactDOM.render(  
  React.createElement(SearchableProductList, { products: [...] }),  
  document.getElementById("root")  
);  
class SearchableProductList extends React.Component {  
  constructor(props) { this.state = { searchText: "" }; ... }  
  render() {  
    return React.createElement("div", null, ...  
      React.createElement(ProductList, { products: this.props.products, searchText: this.state.searchText })  
    );  
  }  
}  
class ProductList extends React.Component { ...  
  render() {  
    var listItems = [];  
    this.props.products.forEach((p) => {  
      var txt = this.props.searchText.toLowerCase();  
      var n = p.name.toLowerCase(); var desc = p.description.toLowerCase();  
      listItems.push(  
        React.createElement(Product, { name: p.name,  
          description: p.description, isBookmarked: p.isBookmarked,  
          display: (txt == "" || n.indexOf(txt) > -1 || desc.indexOf(txt) > -1) })  
      );  
    });  
    return React.createElement("div", null, listItems);  
  }  
}  
class Product extends React.Component { ...  
  render() {  
    return React.createElement("div",  
      { className: this.props.display ? (this.state.isBookmarked ? "box yellow" : "box") :  
        "nodisplay" },  
      React.createElement("h1", null, this.props.name),  
      ... );  
  }  
}
```

```
graph TD
    ReactDOM[ReactDOM.render] --> SearchableProductList[SearchableProductList]
    SearchableProductList --> ProductList[ProductList]
    ProductList --> Product[Product]
    SearchableProductList --> ProductList
    ProductList --> Product
    Product --> Product
```

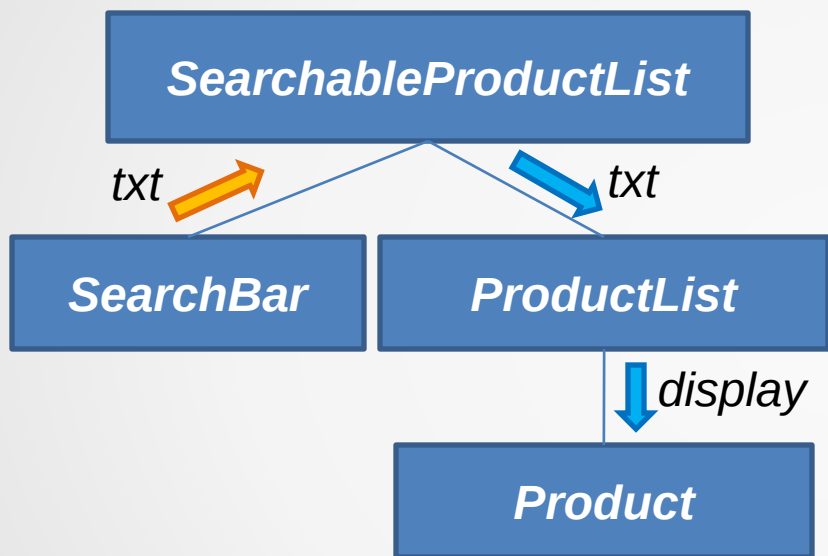
Chuyển dữ liệu ngược lên

- Chuyển dữ liệu từ thành phần con lên thành phần cha
 - Thành phần cha chuyển xuống cho thành phần con một tham chiếu hàm, còn gọi là **hàm gọi lại** (callback function), bằng buộc dữ liệu trên-xuống
 - Thành phần con có thể gọi đến hàm gọi lại của thành phần cha và truyền dữ liệu cho thành phần cha dưới dạng tham số của hàm gọi lại.

Chuyển dữ liệu ngược lên

```
ReactDOM.render(  
  React.createElement(SearchableProductList, { products: [...] }),  
  document.getElementById("root")  
);  
class SearchableProductList extends React.Component {  
  constructor(props) { this.state = { searchText: "" }; ...; this.doSearch = this.doSearch.bind(this); }  
  doSearch(txt) { this.setState({ searchText: txt }); }  
  render() {  
    return React.createElement("div", null,  
      React.createElement(SearchBar, { onSearch: this.doSearch }),  
      React.createElement(ProductList, { products: this.props.products, searchText: this.state.searchText })  
    );  
  }  
}  
class ProductList extends React.Component { ...  
  render() { ... }  
}  
class SearchBar extends React.Component {  
  constructor(props) {  
    super(props);  
    this.handleInputTextChanged = this.handleInputTextChanged.bind(this);  
  }  
  handleInputTextChanged(e) { this.props.onSearch(e.target.value); }  
  render() {  
    return React.createElement("input", { type: "text",  
      placeholder: "Input text to search", onChange: this.handleInputTextChanged  
    });  
  }  
}
```

Kết hợp chuyển dữ liệu lên và xuống



Input text to search

Dell Laptops

Laptops from Dell

Set bookmark

Dell PCs

PCs from Dell

Remove bookmark

HP Laptops

Laptops from HP

Set bookmark

Lap

Dell Laptops

Laptops from Dell

Set bookmark

HP Laptops

Laptops from HP

Set bookmark

JSX

- JSX (**J**ava**S**cript **eX**tension)
 - Cung cấp một cú pháp khác, dễ hơn, thay cho viết lệnh *React.createElement()*.
 - Khai báo các thành phần React tương tự khai báo các đối tượng HTML
 - Ví dụ
 - `<Product name="Dell Laptops" description="Laptops from Dell" isBookmarked="false"/>`
 - ✓ `React.createElement(Product, {name:"Dell Laptops", description:"Laptops from Dell", isBookmarked: false});`
 - `<h1>{this.props.name}</h1>`
 - ✓ `React.createElement("h1", null, this.props.name);`

JSX

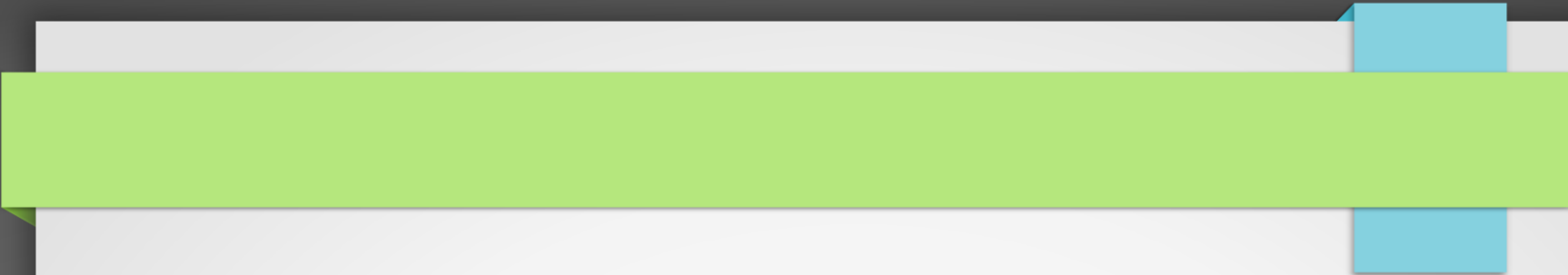
```
class SearchableProductList extends React.Component {  
  ...  
  render() {  
    return <div>  
      <SearchBar onSearch={this.doSearch}/>  
      <ProductList products={this.props.products} searchText={this.state.searchText}/>  
    </div>;  
  }  
}  
ReactDOM.render(  
  <SearchableProductList products={[]} />, document.getElementById('root')  
);
```

Thiết lập môi trường

- Cách 1: Bao hàm thư viện
 - *react.min.js*
 - *react-dom.min.js*
 - *babel.js* nếu sử dụng JSX
 - <https://babeljs.io/>
- Cách 2: Thiết lập IDE
 - "build" ứng dụng để chuyển đổi JSX thành Javascript thuần
 - <https://facebook.github.io/react/docs/installation.html>

Tham khảo

- Bài tập mẫu
 - <http://uet.vnu.edu.vn/~thanhld/lects/webapp-development/react/>



Tiếp theo

Công nghệ web động

