

Bài thực hành số 2: Sử dụng đối tượng state, xử lý sự kiện, sử dụng thư viện Bootstrap

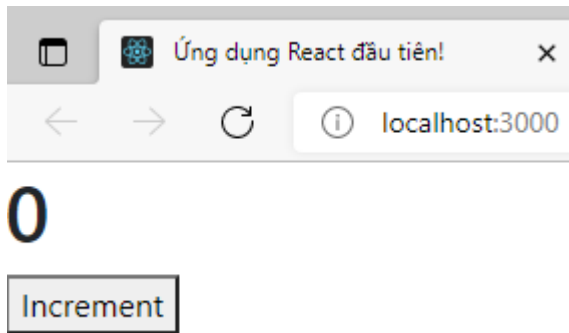
Chúng ta sẽ thay đổi giao diện với những giá trị có thể thay đổi. Lúc này chúng ta dùng state. Viết lại state:

```
state = {  
  count: 0  
};
```

Muốn hiển thị giá trị thuộc tính count của state, chúng ta có thể thay nội dung <h1>

```
<h1>{this.state.count}</h1>
```

Thực thi



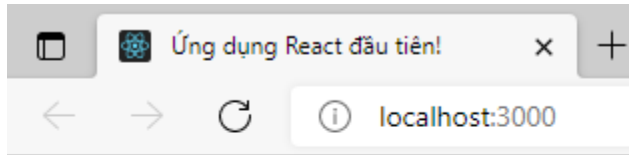
Chúng ta có thể cải thiện một ít tại đây bằng cách thêm một phương thức tên formatCount:

```
formatCount(){  
  const {count} = this.state;  
  return count === 0 ? "Zero" : count;  
}
```

Trong h1 chúng ta thay nội dung:

```
<h1>{this.formatCount()}</h1>
```

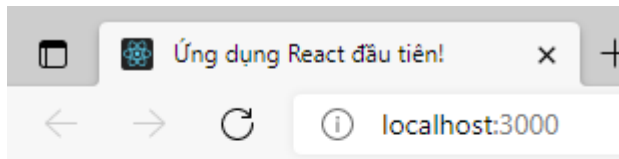
Thực thi



Zero

Increment

Nếu thay đổi giá trị count = 1



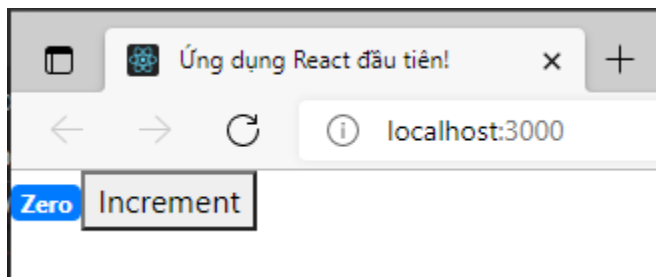
1

Increment

Đổi h1 thành span và thêm thuộc tính className để sử dụng Bootstrap:

```
<span className='badge badge-primary'>{this.formatCount()}</span>
```

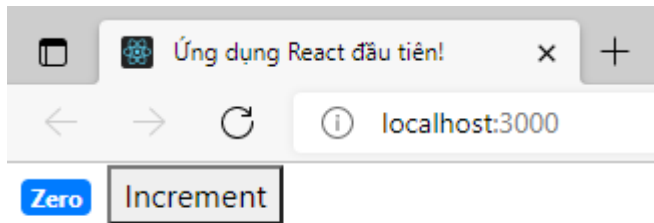
Kết quả



Thay đổi một ít đến className

```
<span className='badge badge-primary m-2'>{this.formatCount()}</span>
```

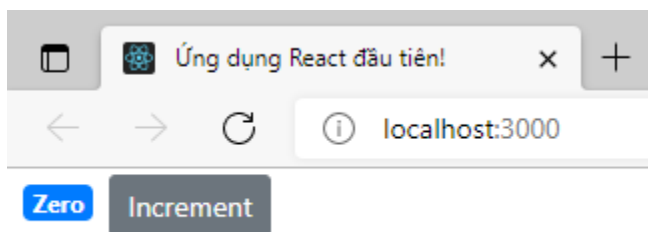
Kết quả



Tương tự thêm className đến button

```
<button className='btn btn-secondary btn-sm'>Increment</button>
```

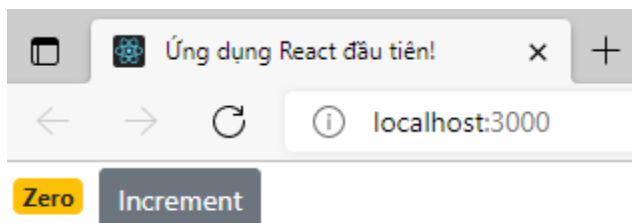
Kết quả



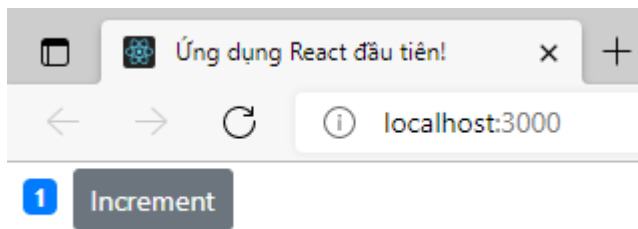
Bây giờ chúng ta cần thực hiện một vài code để tăng tính động bằng cách tạo ra biến classes, khởi tạo giá trị và gán đến giá trị thuộc tính className của span:

```
render() {  
  let classes = "badge m-2 badge-";  
  classes += (this.state.count === 0) ? "warning" : "primary";  
  
  return (  
    <div>  
      <span className={classes}>{this.formatCount()}</span>  
      <button className='btn btn-secondary btn-sm'>Increment</button>  
    </div>  
  );  
}
```

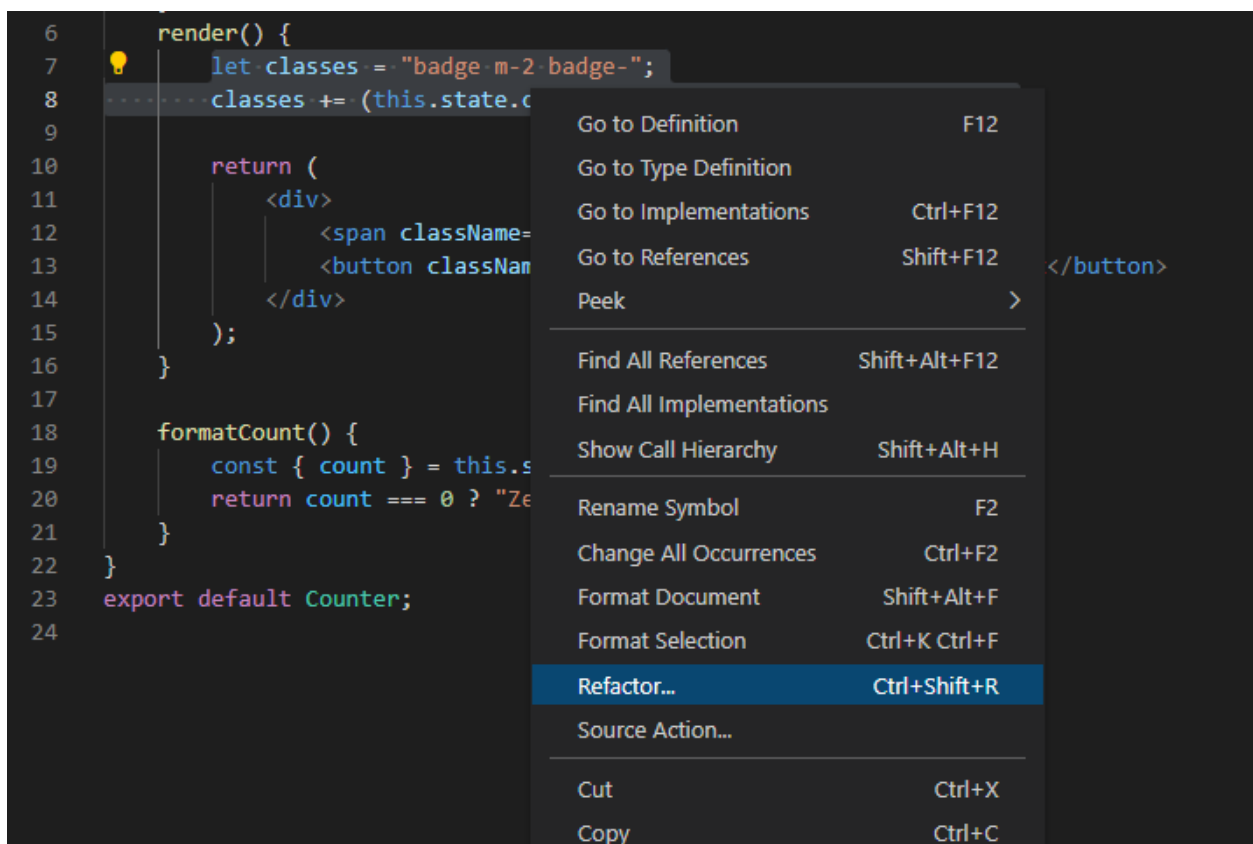
Thực thi với count = 0



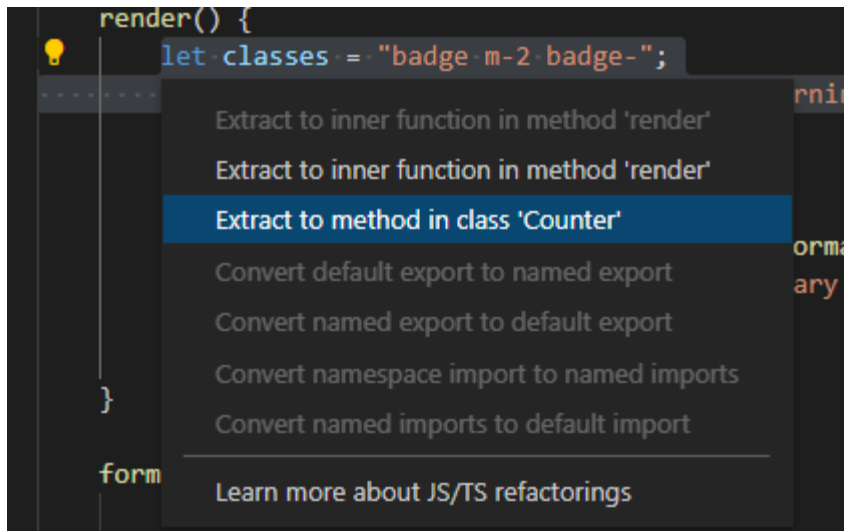
Thực thi với count = 1



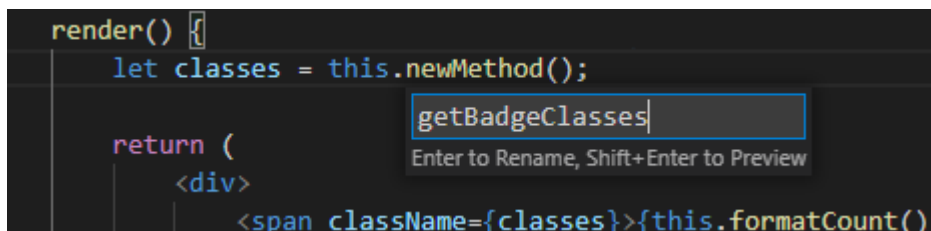
Cải tiến đoạn mã trên (còn gọi là refactoring) bằng cách đặt các dòng mã của biến classes trong một phương thức gọi là getBadgeClasses bằng cách bôi đen toàn bộ dòng code, nhấn chuột phải và chọn Refactor



Chọn Extract to method in class 'Counter'



Thay thế tên mặc định newMethod bằng tên getBadgeClasses và Enter



Thay giá trị thuộc tính className của span bằng phương thức getBadgeClasses. Toàn bộ nội dung của Counter lúc này:

```
import React, { Component } from 'react';  
class Counter extends Component {  
  state = {  
    count: 0  
  };  
  render() {  
    return (  
      <div>  
        <span  
          className={this.getBadgeClasses()}>{this.formatCount()}</span>  
        <button className='btn btn-secondary btn-sm'>Increment</button>  
      </div>  
    );  
  }  
  getBadgeClasses() {  
    let classes = "badge m-2 badge-";  
    classes += (this.state.count === 0) ? "warning" : "primary";  
    return classes;  
  }  
}
```

```

    }

    formatCount() {
      const { count } = this.state;
      return count === 0 ? "Zero" : count;
    }
  }
}
export default Counter;

```

Chúng ta muốn rằng khi nhấn chuột vào button sẽ thực hiện tăng giá trị bằng cách thêm thuộc tính `onClick` và phương thức xử lý sự kiện trong React. Trước tiên chúng ta xây dựng phương thức `handleIncrement()` ngay dưới thuộc tính `state` như sau:

```

state = {
  count: 0
};
handleIncrement = () => {
  this.setState({ count: this.state.count + 1 });
}

```

Kế tiếp thêm thuộc tính `onClick` đến button và gọi phương thức `handleIncrement()`

```

<button onClick={this.handleIncrement} className='btn btn-secondary btn-sm'>Increment</button>

```

Thực thi ứng dụng và nhấn button Increment để tăng giá trị biến Count

