State

Ustun Ozgur

2015-06-13 Cumartesi

State

- Değişken verili uygulamalar
- İki önemli metod: `getİnitialState` ve `setState`
- `setState`: girdiyle şu anki state'i birleştirir
- Ayrıca `replaceState`, şu anki state'i girdi ile değiştirir
- state'e render metodundan şu şekilde erişilir `this.state`

Örnek: Sayaç

```
var Counter = React.createClass({

  getInitialState: function () {
    return {counter: 0};
  },
  increment: function () {
    this.setState({counter: this.state.counter + 1});
  },
  render: function () {
    return <div onClick={this.increment}>
      Counter value is {this.state.counter}
    </div>;

  }

});
```

Genel Bakış

- Bir tane ‘onClick‘ handler var. Bu ‘setState‘’i çağırıyor, birer birer artırarak
- Bu React ile değişken veri (state) yönetiminin özüdür.

State'i Çocuklara Props olarak Geçirmek

- Bir sahip (ana) bileşen (owner component), state'ini çocuklara props olarak geçirebilir.
- Bir bileşenin state'i bir diğerinin props'udur.

Örnek: Sayaçlar: Sayaç Bileşeni

```
var Counter = React.createClass({
  render: function () {
    return <div>
      Counter value is {this.props.counter}
      <button onClick={this.props.click}>Increment</button>
      </div>
  }

});
```

Örnek: Sayaçlar Bileşeni

```
var Counters = React.createClass({
  getInitialState: function () {
    return {counter1: 10, counter2: 0};  },
  incrementCounter1: function () {
    this.setState({counter1: this.state.counter1 + 1,
                   counter2: this.state.counter2 - 1})  },
  incrementCounter2: function () {
    this.setState({counter1: this.state.counter1 - 1,
                   counter2: this.state.counter2 + 1})  },
  render: function () {
    return <div>
      <Counter click={this.incrementCounter1}
       counter={this.state.counter1}/>
      <Counter click={this.incrementCounter2}
       counter={this.state.counter2}/>
      </div>  }})
```

bind kullanan daha kısa versiyonu

```
var Counters = React.createClass({
  getInitialState: function () {
    return {counter1: 10, counter2: 0};    },
  incrementCounter: function (increment) {
    this.setState({
      counter1: this.state.counter1 + increment,
      counter2: this.state.counter2 - increment})    },
  render: function () {
    return <div>
      <Counter
       click={this.incrementCounter.bind(this, 1)}
       counter={this.state.counter1}/>
      <Counter
       click={this.incrementCounter.bind(this, -1)}
       counter={this.state.counter2}/>
      </div>    }})
```

Özet

- Çocuk bileşenler hiçbir zaman props'ları doğrudan değiştirmez.
- Ana bileşene bir olay olduğunu props olarak geçirilen fonksiyonlar ile haber verirler.
- Ana bileşen state'i değiştirir.
- React state'i aşağı doğru props olarak aktarır.
- Her şey yeni baştan görüntülenir (render edilir).

Alıştırma:

- Todo uygulamasını liste öğeleri state'te olacak şekilde değiştirin.
- Todo bileşenlerinin tamamlanmış/tamamlanmamış özelliğini tamamlayın.
- İpucu: todo elemanları bir string listesi gibi mi tutulmalı, bir nesne listesi olarak mı?
- Bitmiş/bitmemiş şeklindeki filtreleri tamamlayın.
- İpucu: Filtrenin şu an aktif olup olmadığını bir state değişkeni ile izleyin.
- Bir todo elemanının listeden çıkarılmasını tamamlayın.
- Kalan eleman sayısının gösterilmesi özelliğini gerçekleştirin.