

# React ile Formlar

Ustun Ozgur

2015-06-13 Cumartesi

- KontROLSÜZ (Uncontrolled)
- KONTROLLÜ (Controlled)

# Kontrolsüz (Uncontrolled)

- Önerilmez
- Ana DOM bileşenleri veriyi saklar, state değil
- Değerler olaylardan (event) ya da ref'ler üzerinden DOM elemanlarından okunur
- ref: bir çocuk bir eşsiz bir gösterici (unique identifier)
- `this.refs` aynı props, state gibi ref'lerden oluşan bir nesnedir

## Örnek

```
<div>  
  <span ref="inner">This is the inner text</span>  
</div>
```

- İçteki span elemanına şu şekilde ulaşılabilir `this.refs.inner`
- Gerçek bir DOM düğümüne ulaşmak için, sunlar kullanılabilir:
  - `React.findDOMNode`: tercih edilen
  - `this.refs.inner.getDOMNode`: eski yöntem
- Değeri almak için DOM düğümüne eriştikten sonra `value` değerini okuyun
- `this.refs.inner.getDOMNode().value`

## Örnek: Kontrolsüz Girdili (Input) Bir Form

```
<form onSubmit={this.onSubmit}>
```

```
<input ref={name} placeholder="What is your name"/>
```

```
<input ref={surname} placeholder="What is your surname?"/>
```

```
<button type="submit">Submit</button>
```

```
</form>
```

## Örnek (devam)

```
var MyForm = React.createClass({
  render: function () {

    return <form onSubmit={this.onSubmit}>
      <input ref="name" placeholder="What is your name"/>
      <input ref="surname"
        placeholder="What is your surname?"/>
      <textarea ref="comment" placeholder="Comment"/>
      <label><input ref="fromDenmark"
        type="checkbox"/>Are you from Denmark?</label>
      <label>Gender:
        <select ref="gender">
          <option>Male</option>
          <option>Female</option>
        </select></label>
      <button type="submit">Submit</button></form>
```

## Örnek (devam)

```
onSubmit: function (e) {  
  e.preventDefault();  
  var name = React.findDOMNode(this.refs.name).value;  
  var surname = React  
    .findDOMNode(this.refs.surname).value;  
  var comment = React  
    .findDOMNode(this.refs.comment).value;  
  var fromDenmark = React  
    .findDOMNode(this.refs.fromDenmark).checked;  
  var gender = React.findDOMNode(this.refs.gender).value;  
  
  console.log("Form submitted with",  
    name, surname, comment,  
    fromDenmark, gender);  
  
});
```

- İşaret kutuları için (checkbox), value yerine checked özelliğini kullanılır
- textarea bileşeninin çocuk elemanları yoktur, HTML'de vardır. textarea için yine value değeri kullanılır.
- defaultValue : varsayılan (ilk )değer için bu kullanılır.
- Formların bir de onSubmit adında bir olay fonksiyonu vardır, ama bu aynı zamanda formun asıl metodunu (GET/POST) çalıştırır, bu nedenle preventDefault kullanarak asıl metodun çağrılması engellenir.
- Buradaki olaylar (Event) sentetik olay adı verilen React'e özel olaylardır ve ana DOM olaylarını normalize ederler.



- Tercih edilen yöntem
- Her girdi bileşeninin bir değer ya da işaretlenmiş özelliği vardır
- Her girdi bileşeni bir onChange olayı yaratır.
- State'i bu onChange olayında değiştiririz.

## Örnek: Kontrollü Bir girdi (ilk değer)

```
var MyForm = React.createClass({  
  
  getInitialState: function () {  
    return {name: ''};  
  },  
  render: function () {  
    return <form onSubmit={this.onSubmit}>  
      <input value={this.state.name} placeholder="What is your name?" />  
    </form>  
  }  
});
```

## Örnek (devam) (olay fonksiyonları)

```
var MyForm = React.createClass({  
  
  getInitialState: function () {  
    return {name: ''};  
  },  
  changeName: function (e) {  
    this.setState({name: e.target.value});  
  },  
  render: function () {  
    return <form onSubmit={this.onSubmit}>  
      <input onChange={this.changeName}  
        value={this.state.name}  
        placeholder="What is your name?"/>  
    </form>}})
```

## Örnek (submit (gönder) metodu)

```
var MyForm = React.createClass({
  onSubmit: function () {
    console.log("the form values are", this.state.name); },
  getInitialState: function () {
    return {name: ''}; },
  changeName: function (e) {
    this.setState({name: e.target.value}); },
  render: function () {
    return <form onSubmit={this.onSubmit}>
      <input onChange={this.changeName}
        value={this.state.name}
        placeholder="What is your name?"/>
    </form>  {this.props.children})
```

# Formlara Yönelik Diğer Olaylar

- onBlur
- onFocus

# Hata Ayıklama (Debugging) için bir İpucu

- State'i görsel olarak öldüğü gibi ekrana yazdırma
- `JSON.stringify(this.state, null, 4)` kullanarak state'i yazdırabiliriz.

## Örnek:

```
var MyForm = React.createClass({
  onSubmit: function () {
    console.log("the form values are", this.state.name); },
  getInitialState: function () {
    return {name: ''}; },
  changeName: function (e) {
    this.setState({name: e.target.value}); },
  render: function () {
    return <form onSubmit={this.onSubmit}>
      <input onChange={this.changeName}
        value={this.state.name}
        placeholder="What is your name?"/>
      <pre>{JSON.stringify(this.state, null, 4)}</pre>
    </form>  })
```

## Alıştırma 1/3:

- Kullanıcının sadece sayı girebildiği bir kredi kartı form input'u ekleyin.

İpucu: Bir değişkenin sadece sayılardan oluştuğunu anlamak için su regex'i kullanabilirsiniz `/^\d+$/`. `test(foo)`

- Birkaç sayı girdikten sonra tamamen silebiliyor musunuz? Silemiyorsanız neyi değiştirmeniz gerekli? Neden? İpucu: Ya regex'i değiştirin ya da başka bir yöntem bulun.
- Tireleri girecek şekilde regex'i değiştirin.



## Alıştırma 2/3:

- Kullanıcı ad alanına giriş yaparken onları cinsiyetlerine göre Bey/Hanım, olarak selamlayın. Örneğin adını Ali olarak girerse Merhaba Ali Bey desin, Ayşe olarak girerse Merhaba Ayşe Hanım desin. Erkek adları: ["Ali", "Ahmet"]. Kadın adları ["Ayşe", "Fatma"].
- Bunun dışındaki adlara sadece adıyla hitap edin. Merhaba Hüseyin.
- Eğer ad yoksa, ekranda Merhaba yazmasın, ekran boş olsun.

## Alıştırma 3/3:

- Kullanıcı ad alanından kart alanına geçerken eğer 3 harften kısa bir ad yazdıysa bir uyarı oluşturun. Ancak daha alanı terk etmediyse bu uyarı görüntülenmemeli.

İpucu: Takip edilmesi gereken state değişkenleri hakkında düşünün. Bir değişkenin geçerli olup olmadığı bilgisi state'te tutulmalı mı? Artıları ve eksikleri düşünün.

- Submit tuşuna tıklayınca bir validasyon ekleyin. İsim en az üç harf, kredi kartı tireler hariç en az 16 rakam olmalı. Formun validasyon değerini state'te tutmalı mıyız? Bu konuya kafa yörün.
- Todo uygulamasında, yeni bir todo ekleme özelliğini ekleyin.