

# React ile AJAX

Ustun Ozgur

2015-06-13 Cumartesi

# React ile AJAX

- React, Ajax'tan tamamen bağımsızdır
- Sadece bir görüntü katmanıdır
- AJAX isteğini yapın, örneğin jQuery get metodu ile
- Çıktı verisini bileşenin state'inde kaydedin
  - Örneğin success fonksiyonunda setState kullanarak.

# componentDidMount

- Bu da render gib bir özel yaşamdongusu metodudur
- Bileşen gerçek DOM'e monte edildiğinde çağrılır
- Teknik: Bileşen monte olmadan önce state'i başlatın. Monte olduktan sonra AJAX isteğini başlatın.
- AJAX sonucu dönünce, bileşen hala monte durumda ise state'i güncelleyin.

# Örnek: Github

- Amacımız bir kullanıcının depo listesini alıp görüntülemek.
- Github'ın bir ucnoktası var  
`https://api.github.com/users/USER_NAME/repos`  
(buradaki USER\_NAME yerine gerçek bir kullanıcı adı gelecek)
- Bu bir depo dizisi halinde bir json çıktısı döner.

## getInitialState kullanarak boş depo dizisi

```
var Repos = React.createClass({
  getInitialState: function () {
    return {repos: []};
  },
  render: function () {
    var renderRepo = function (repo) {
      return <div>Repo {repo.name} {repo.date}</div>
    }
    return <div>
      {this.state.repos.map(renderRepo)}
    </div>

  }

});
```

# Github'dan veriyi çekmek

```
var GITHUB_ENDPOINT = "https://api.github.com/users/ustun/repos";
var Repos = React.createClass({
  getInitialState: function () { return {repos: []};},

  componentDidMount: function () {
    $.getJSON(GITHUB_ENDPOINT, function (data) {
      this.setState({repos: data});
    }.bind(this));
  },
  render: function () {
    var renderRepo = function (repo) {
      return <div>Repo {repo.name} {repo.created_at}</div>;
    }
    return <div>{this.state.repos.map(renderRepo)}</div>
  }
});
```

# Bileşenin hala monte olduğundan emin olmak

- İsteği başlattığımızda bileşen monte
- AJAX çağrısı dönünce emin değiliz.
- React şu an monte edilmemiş olan bir bileşenin state'ini değiştiriyor olabilir.
- Bunu önlemek için state'i monte etmeden önce `isMounted` kullanın.

```
componentDidMount: function () {  
  $.getJSON(GITHUB_ENDPOINT, function (data) {  
    if (this.isMounted()) {  
      this.setState({repos: data});  
    }  
  }.bind(this)); },
```

- AJAX sonuçlarında setState kullanın.
- AJAX çağrıları genelde componentDidMount metodunda başlatılır.
- AJAX çağrıları bir olay fonksiyonunda (event handler) da çağrılabilir.



- Instagram'ın şu adreste bir API'i var
- `https://api.instagram.com/v1/tags/nofilter/media/recent?client_id=CLIENT_ID` nofilter herhangi etiketin ismi.
- Yani `https://api.instagram.com/v1/tags/TAG_NAME/media/recent?client_id=CLIENT_ID` TAGNAME doldurulacak
- Workshop için `https://\T1\instagram.com/developer/` adresinden bir Instagram uygulaması oluşturun ve oradaki client id'yi kullanın.
- HDR etiketli instagramları getirecek bir uygulama yapın.
- API ilk birkaç sonucu döner, ancak bir de next adında bir parametresi vardır. Bunu kullanarak bir "Daha Fazla Sonuç Getir" düğmesi yapın.
- Uyarı (sonraki sayfa)

- Instagram same origin policy kullanmaktadır. Bu alıştırma için bunu devre dışı bırakın
- `open -a Google\ Chrome --args --disable-web-security`
- `chrome.exe --disable-web-security`