

BMB315. Web Programlama

Ders 5:

Front-end Frameworks: AppML, AngularJS, Angular,
React, vuejs

AppML (Application Modeling Language)

- AppML bir uygulama modelleme dilidir.
- AppML herhangi bir web sayfasında çalışabilir ve ek bir yükleme gerektirmez.
- AppML HTML uygulamasına veri almak için kullanılır. Bu veri:
 - Bir nesne
 - Bir dosya
 - Bir veritabanıolabilir.

<https://www.w3schools.com/appml/default.asp>

AppML Örneği 1

```
<!DOCTYPE html>
<html lang="en-US">
<title>Customers</title>
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
<script src="http://www.w3schools.com/appml/2.0.2/appml.js"></script>
<body>
  <div class="container" appml-data="dataObj">
    <h1>Customers</h1>
    <table class="table table-striped table-bordered">
      <tr>
        <th>Customer</th>
        <th>City</th>
        <th>Country</th>
      </tr>
      <tr appml-repeat="records">
        <td>{{CustomerName}}</td>
        <td>{{City}}</td>
        <td>{{Country}}</td>
      </tr>
    </table>
  </div>
  <script>
    var dataObj = {
      "records": [
        { "CustomerName": "Alfreds Futterkiste", "City": "Berlin", "Country": "Germany" },
        { "CustomerName": "Ana Trujillo Emparedados y helados", "City": "México D.F.", "Country": "Mexico" },
        { "CustomerName": "Cactus Comidas para llevar", "City": "Buenos Aires", "Country": "Argentina" },
        { "CustomerName": "Centro comercial Moctezuma", "City": "México D.F.", "Country": "Mexico" },
        { "CustomerName": "Chop-suey Chinese", "City": "Bern", "Country": "Switzerland" },
        { "CustomerName": "Comércio Mineiro", "City": "São Paulo", "Country": "Brazil" }
      ]
    };
  </script>
</body>
</html>
```

BootStrap

appml.js

Çift süslü parantez
AppML satırıdır.
Burada veri istenir.
{{alan_ismi}}

Object veri

JSON Veri

Hiç döngü kurmadan kolay bir şekilde veriyi tablo içine yazmamızı sağladı.

AppML Örneği 2

```
<!DOCTYPE html>
<html lang="en-US">
<title>Customers</title>
<link rel="stylesheet" href="style.css">
<script src="http://www.w3schools.com/appml/2.0.2/appml.js"></script>
<body>

  <p id="demo"></p>

  <script>
    var myObj, myArr, len, i, txt = "";

    // Create an AppML object and fetch the data
    myObj = new AppML();
    myObj.dataSource = "http://www.w3schools.com/appml/customers.php";
    myObj.getData();

    // Locate the data records
    myArr = myObj.data.records;
    len = myArr.length;

    // Display the records
    for (i = 0; i < len; i++) {
      txt += myArr[i].CustomerName + "<br>";
    }
    document.getElementById("demo").innerHTML = txt;
  </script>

</body>
</html>
```

Otomatik çekme yerine
AppML sınıfı üzerinden
veri çekme

Nesne oluştur

Nesneye veri kaynağını
belirle
Sunucu tarafı!!! Önümüzdeki
haftalarda

Verileri sunucudan getir

Kayıtları diziye at

Verideki kayıt uzunluğu

Bu klasik süreçtir. AppML {{alan_adi}}
kullanımı bizi bu zahmetli koddan kurtarır.

AppML Örneği 3

- Sunucudan gelen veri içinde ek bilgilere göre sayfalama işlemi yapmak çok kolaydır.

```
{
  "user" : 0, "recPos" : 1, "fromRec" : 1, "toRec" : 7, "totalRecCounter" : 26, "records" : [{"Artist" : "Bob Dylan", "Title" : "Empire Burlesque", "Country" : "USA"}, {"Artist" : "Bonnie Tyler", "Title" : "Hide your heart", "Country" : "UK"}, {"Artist" : "Dolly Parton", "Title" : "Greatest Hits", "Country" : "USA"}, {"Artist" : "Gary Moore", "Title" : "Still got the blues", "Country" : "UK"}, {"Artist" : "Eros Ramazzotti", "Title" : "Eros", "Country" : "EU"}, {"Artist" : "Bee Gees", "Title" : "One night only", "Country" : "UK"}, {"Artist" : "Dr. Hook", "Title" : "Sylvias Mother", "Country" : "UK"}],
  "rowsperpage" : 7,
  "data" : {
    "type" : "xmlfile",
    "filename" : "cd_catalog.xml",
    "record" : "CD",
    "items" : [
      { "name" : "Artist", "nodename" : "ARTIST" },
      { "name" : "Title", "nodename" : "TITLE" },
      { "name" : "Country", "nodename" : "COUNTRY" }
    ]
  }
}
```

JSON Veri

```
<!DOCTYPE html>
<html lang="en-US">
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
<script src="http://www.w3schools.com/appml/2.0.2/appml.js"></script>
<body>
  <div class="container" appml-data="appml.php?model=model_cd_from_xml">
    <h1>CD Collection</h1>
    <h3>Extracted from an XML file</h3>
    <div appml-include-html="inc_listcommands_nofilter.htm"></div>
    <table class="table table-striped table-bordered">
      <tr>
        <th>Title</th>
        <th>Artist</th>
        <th>Country</th>
      </tr>
      <tr appml-repeat="records">
        <td>{{Title}}</td>
        <td>{{Artist}}</td>
        <td>{{Country}}</td>
      </tr>
    </table>
  </div>
</body>
</html>
```

CD Collection

Extracted from an XML file

1 - 7 of 26

Title	Artist	Country
Empire Burlesque	Bob Dylan	USA
Hide your heart	Bonnie Tyler	UK
Greatest Hits	Dolly Parton	USA
Still got the blues	Gary Moore	UK
Eros	Eros Ramazzotti	EU
One night only	Bee Gees	UK
Sylvias Mother	Dr. Hook	UK

AngularJS

- AngularJS, HTML yeni özellikler ile genişletir.
- AngularJS, tek sayfalı uygulamalar (Single Page Applications) için çok iyi bir çözümdür.

The diagram illustrates the AngularJS data binding mechanism. A red arrow points from the text 'AngularJS kütüphanesi' to the `<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>` tag in the HTML code. Another red arrow points from the `ng-model="name"` attribute in the `<input type="text" ng-model="name" placeholder="Enter name here">` tag to the `{{name}}` expression in the `<h1>Hello {{name}}</h1>` tag. A third red arrow points from the `{{name}}` expression to the text 'Metin kutusundaki name değıştikçe H1 etiketi içindeki name'de değışir. AppML gibi veri çekme çift süslü parantez ile yapılır.'

```
<!DOCTYPE html>
<html>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
<body>
  <div ng-app="">
    <p>Input something in the input box:</p>
    <p>Name : <input type="text" ng-model="name" placeholder="Enter name here"></p>
    <h1>Hello {{name}}</h1>
  </div>
</body>
</html>
```

Metin kutusundaki name değıştikçe H1 etiketi içindeki name'de değışir. AppML gibi veri çekme çift süslü parantez ile yapılır.

AngularJS

- data-ng-init başlangıç veri etiketidir. Bir firstName değişkeni içinde veri vardır.
- data-ng-bind ile başlangıçta verilen veri çekilebilir. Ayrıca verinin bulunduğu değişken özelliği kullanılabilir.

```
<div data-ng-app="" data-ng-init="firstName='John'">
  <p>The name is <span data-ng-bind="firstName"></span></p>
</div>
```

- Süslü parantezler arasında direkt işlem dahi yaptırılabilir.

```
<div ng-app="">
  <p>My first expression: {{ 5 + 5 }}</p>
</div>
```

```
<div ng-app="" ng-init="quantity=1;cost=5">
  <p>Total in dollar: {{ quantity * cost }}</p>
</div>
```

- ng-app, bir AngularJS uygulamasının başlatır.
- ng-init, uygulama verisini başlatır.
- ng-model, uygulama verisine HTML control (input, select, textarea) değerlerini bağlar.

AngularJS

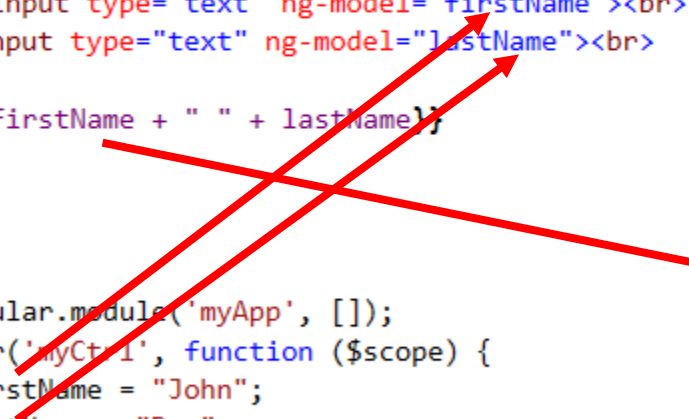
```
<p>Try to change the names.</p>

<div ng-app="myApp" ng-controller="myCtrl">

  First Name: <input type="text" ng-model="firstName"><br>
  Last Name: <input type="text" ng-model="lastName"><br>
  <br>
  Full Name: {{firstName + " " + lastName}}

</div>

<script>
  var app = angular.module('myApp', []);
  app.controller('myCtrl', function ($scope) {
    $scope.firstName = "John";
    $scope.lastName = "Doe";
  });
</script>
```



Metin kutularındaki değişimden direkt etkilenir. İki değişkenin birleştirilmesine dikkat.

ng-controller bir JavaScript Nesnesidir ve JavaScript bölümü ile iletişimde kullanılır.

AngularJS

- ng-repeat foreach'e benzer bir döngü ile dizi veya veri kümesi çekilebilir.

```
<div ng-app="" ng-init="names=[
  {name:'Jani',country:'Norway'},
  {name:'Hege',country:'Sweden'},
  {name:'Kai',country:'Denmark'}]">

  <ul>
    <li ng-repeat="x in names">
      {{ x.name + ', ' + x.country }}
    </li>
  </ul>

</div>
```

Angular JS

- AngularJS Filters: Verileri formatlı biçimde gösterme
- AngularJS HTML: Sunucudan veri çekme
- AngularJS Tables: Table özelliğini verilerle kullanma
- AngularJS Events: Angular olay yönetimi
- AngularJS Forms: Form içindeki elementleri ayarlama
- Angular Validation: Doğrulama işlemi

AngularJS vs. Angular

- AngularJS, Javascript tabanlı açık kaynaklı bir front-end framework'üdür. Tek sayfalı uygulamalar geliştirmek için ortaya çıkmıştır. En son sürümü 1.8'dir.
- Angular, AngularJS'ye tamamen yeni bir yapıya taşınmış halidir. Sürüm 2.0'dan sonra köklü bir değişikliğe gidilmiştir. En son sürümü 9'dur.

AngularJS vs. Angular

- Mimari
 - AngularJS, MVC, Model View Controller Design üzerinde çalışır. Burada view, modelde bulunan verileri gösterir ve verileri controller üzerinde işler.
 - Angular, bileşenleri (component) ve yönergeleri (directives) kullanır. Burada bileşen, bir şablon (template) içeren yönerge dir.
- Dil
 - AngularJS kodu Javascript ile yazılmıştır.
 - Angular, Typescript kullanır. TypeScript, JavaScript'in üst kümesidir ve geliştirme işlemi sırasında statik yazım sağlar. TypeScript, türler ekleyerek JavaScript'i genişletir. TypeScript, Microsoft tarafından geliştirilmiş ve geliştirilmeye devam etmektedir. TypeScript web sitesinde ön plana çıkan özelliği, hataları yakalama ve kodu çalıştırmadan önce düzeltmeler sunma konusunda size zaman kazandırır. (<https://www.typescriptlang.org/>) (Google tarafından desteklenen bir ürünün Microsoft tarafından desteklenen bir yapıyı kullandığını görüyoruz.)
- Mobil
 - AngularJS kodu mobil uyumlu değildir.
 - Angular geliştirilen uygulamalar mobil tarayıcı dostudur.
- İfade söz dizimi (Syntax)
 - AngularJS, `{{}}`, verileri görünüm ve model arasında bağlamak için kullanılır. Özel yöntemler, ng-bind de aynısını yapmak için kullanılabilir.
 - Veri bağlamaya (data binding) gelince, Angular, AngularJS'den daha sezgiseldir. Bir AngularJS geliştiricisi, bir özelliği veya bir olayı bağlamak için düzeltme yönergesini hatırlamalıdır. Angular durumunda, dil olay bağlama (event binding) için `()` ve özellik bağlama (property binding) için `[]` kullanır.

AngularJS vs. Angular

- Dependency Injection (DI)
 - Hem AngularJS hem de Angular DI kullanımı, ancak yaptıkları tamamen farklıdır.
 - AngularJS'de DI, çeşitli link fonksiyonları, controller fonksiyonları ve directive tanımlarına eklenmiştir.
 - Angular directives, constructor fonksiyonları ve sağlayıcıları (providers) kullanarak hiyerarşik bir DI sistemi kullanır.
 - (Daha fazla Bilgi için SOLID ve son harf Dependency Injection konusuna bakın.)
- Command-Line Interface (CLI)
 - AngularJS'te yoktur.
 - Component'ler ve servisler üretmek ve projeleri hızlı ve verimli bir şekilde oluşturmak için kullanılır. Dinamik tip kontrolü ve benzer özellikler ile farklı platformlar için aynı projenin farklı sürümlerini kolayca oluşturabilirsiniz.
- Routing
 - AngularJS: Routing @routeProvider.when, daha sonra yönlendirme bilgisi sağlamak için kullanılır.
 - Angular: @Route yapılandırması, yönlendirme bilgilerini tanımlamak için kullanılır.
- Yönetim
 - Yönetim AngularJS projesinin, kaynak kodunun boyutu arttıkça yönetilmesi zordur.
 - Angular kod daha iyi yapılandırılmıştır, daha büyük uygulamaları oluşturmak ve yönetmek kolaydır.

React

- <https://tr.reactjs.org/>
- <https://erdincuzun.com/react-native/01-react/>
- <https://www.w3schools.com/react/default.asp>
- <https://erdincuzun.com/react-native/03-javascript-ve-es6/>

Vue.JS

- <https://vuejs.org/>
- <https://vuejs.org/v2/guide/>

Kaynaklar

- AppML
 - <https://www.w3schools.com/appml/default.asp>
- AngularJS ve Angular
 - <https://www.tutorialspoint.com/difference-between-angularjs-and-angular>
 - <https://www.w3schools.com/angular/default.asp>
 - <https://angular.io/>
 - <https://angularjs.org/>
- React
 - <https://erdincuzun.com/react-native/01-react/>
 - <https://www.w3schools.com/react/default.asp>
 - <https://erdincuzun.com/react-native/03-javascript-ve-es6/>
- Vue.JS
 - <https://vuejs.org/v2/guide/>
 - <https://www.tutorialspoint.com/vuejs/index.htm>