



ANGULARJS

by Google

EntwicklerCamp
13. April 2016

Innovative Software-Lösungen.

www.assono.de

Bernd Hort

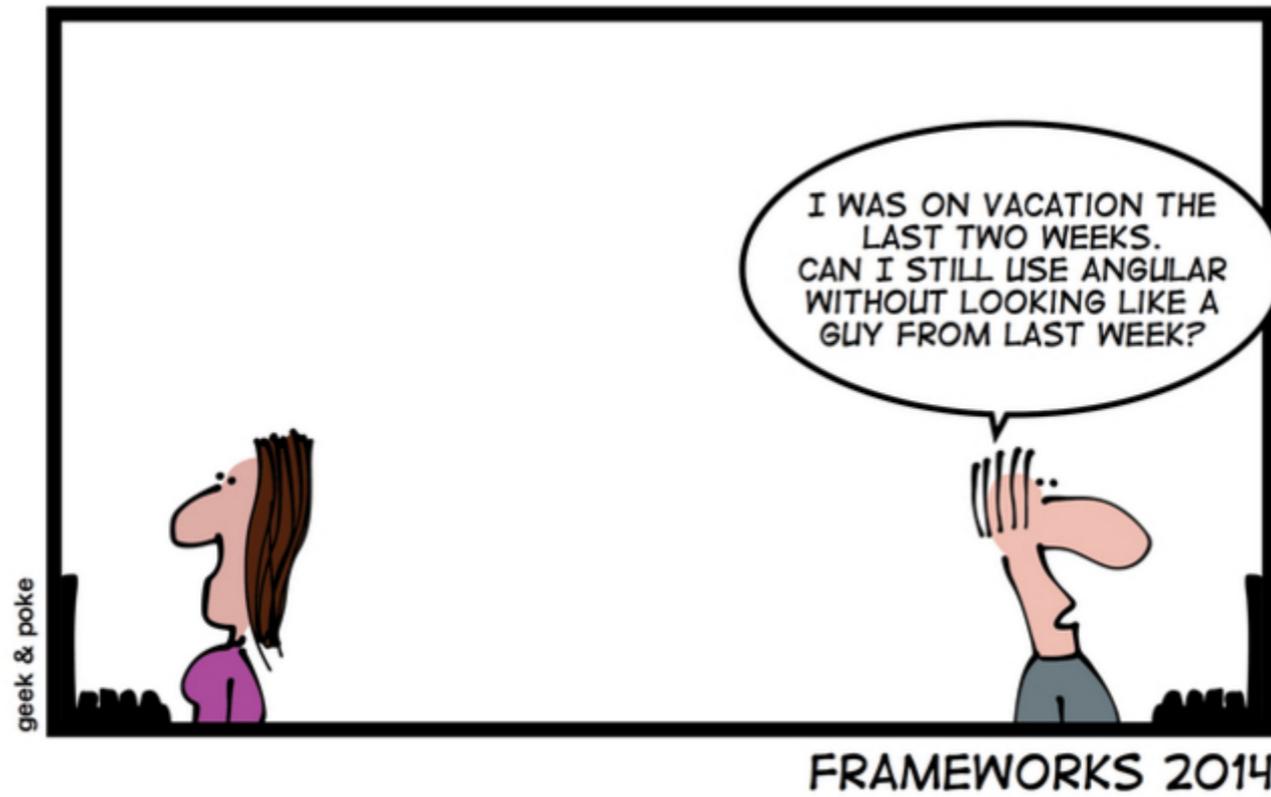
Diplom-Informatiker, Universität Hamburg
seit 1995 entwickle ich Lotus Notes
Anwendungen

IBM Certified Application Developer,
System Administrator & Instructor

Sprecher auf diversen Konferenzen &
Lotusphere 2008/IBM Connect 2014/IBM ConnectED 2015/
IBM Connect 2016



 bhort@assono.de
 <http://www.assono.de/blog>
 040/73 44 28-315



 Download AngularJS 1



(1.5.3 / 1.2.29)

Try the new Angular 2

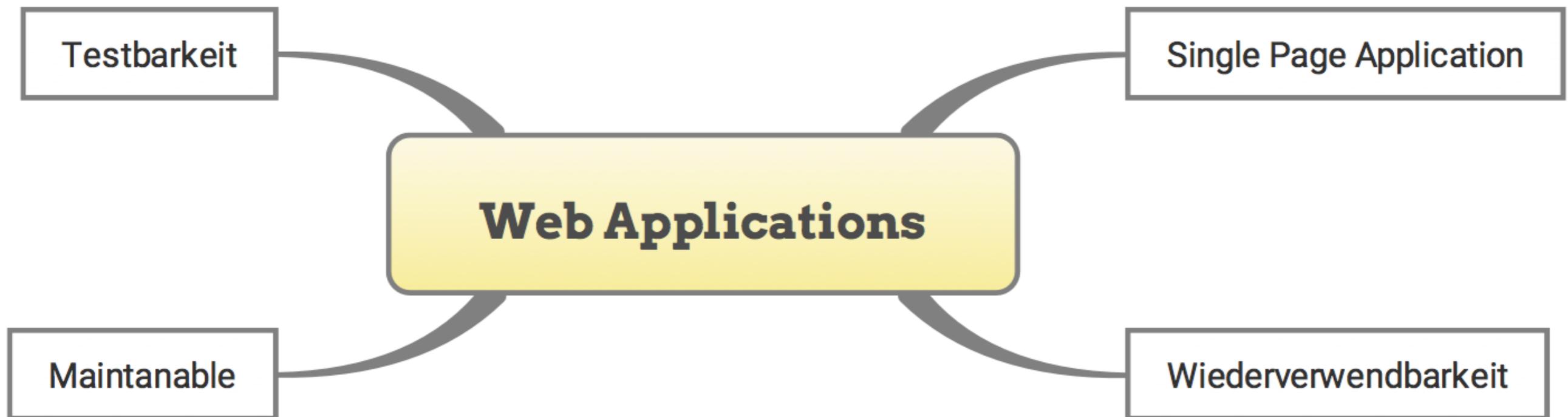


2.0 NOW IN BETA!

Agenda

- Einführung
- Directives
- Model-View-Controller
- Expressions
- Data Binding
- Single Page Application - SPA
- Form Management
- Server Kommunikation
- Fragen und Antworten

Motivation



```

<body ng-app="app">
<div class="container" ng-controller="GuessTheNumberController">
  <h2>Guess the number !</h2>
  <p class="text-info">Guess the computer generated random number between 1 and 1000.</p>
  <label>Your Guess: <input type="number" ng-model="guess"/>
  <button ng-click="verifyGuess()" class="btn btn-primary btn-sm">Verify</button>
  <button ng-click="restartGame()" class="btn btn-warning btn-sm">Restart</button>
  <p>
    <p ng-show="deviation > 0" class="alert alert-warning">Your guess is higher.</p>
    <p ng-show="deviation < 0" class="alert alert-warning">Your guess is lower.</p>
    <p ng-show="deviation === 0" class="alert alert-success">Yes! That's it.</p>
  </p>
  <p class="text-info">No of guesses : <span class="badge">{{noOfTries}}</span><p>
</div>
<script src="../bower_components/angular/angular.js"></script>
<script type="text/javascript">
  angular.module('app', [])
    .controller('GuessTheNumberController', GuessTheNumberController);
  function GuessTheNumberController($scope) {
    $scope.verifyGuess = function () {
      $scope.deviation = $scope.original - $scope.guess;
      $scope.noOfTries = $scope.noOfTries + 1;
    }
    $scope.initializeGame=function() {
      $scope.noOfTries = 0;
      $scope.original = Math.floor(Math.random() * 1000) + 1);
      $scope.guess = null;
      $scope.deviation = null;
    }
    $scope.initializeGame();
  }
</script>
</body>

```

Direktiven / Directives

- HTML-Elemente, die von AngularJS mit zusätzlichen Fähigkeiten ausgestattet werden
- Direktiven können Tagnamen, Attribute, Kommentare oder CSS-Klassennamen sein
 - Bevorzugt werden Tagnamen und Attribute

```
<my-dir></my-dir>  
<span my-dir="exp"></span>  
<!-- directive: my-dir exp -->  
<span class="my-dir: exp;"></span>
```

- Standard oder selbst definierte Direktiven

AngularJS-Applikation

```
<body ng-app="app">
<div class="container" ng-controller="GuessTheNumberController">
  <h2>Guess the Number !</h2>
  <p class="well lead">Guess the computer generated random number between 1 and 1000.</p>
  <label>Your Guess: </label><input type="number" ng-model="guess"/>
  <button ng-click="verifyGuess()" class="btn btn-primary btn-sm">Verify</button>
  <button ng-click="initializeGame()" class="btn btn-warning btn-sm">Restart</button>
  <p>
    <p ng-show="deviation<0" class="alert alert-warning">Your guess is higher.</p>
    <p ng-show="deviation>0" class="alert alert-warning">Your guess is lower.</p>
    <p ng-show="deviation===0" class="alert alert-success">Yes! That's it.</p>
  </p>
  <p class="text-info">No of guesses : <span class="badge">{{noOfTries}}</span><p>
</div>
<script src="../bower_components/angular/angular.js"></script>
<script type="text/javascript">
  angular.module('app', [])
    .controller('GuessTheNumberController', function($scope) {
      function GuessTheNumberController($scope) {
        $scope.verifyGuess = function () {
          $scope.deviation = $scope.original - $scope.guess;
          $scope.noOfTries = $scope.noOfTries + 1;
        }
        $scope.initializeGame=function() {
          $scope.noOfTries = 0;
          $scope.original = Math.floor(Math.random() * 1000) + 1);
          $scope.guess = null;
          $scope.deviation = null;
        }
        $scope.initializeGame();
      }
    }
  </script>
</body>
```

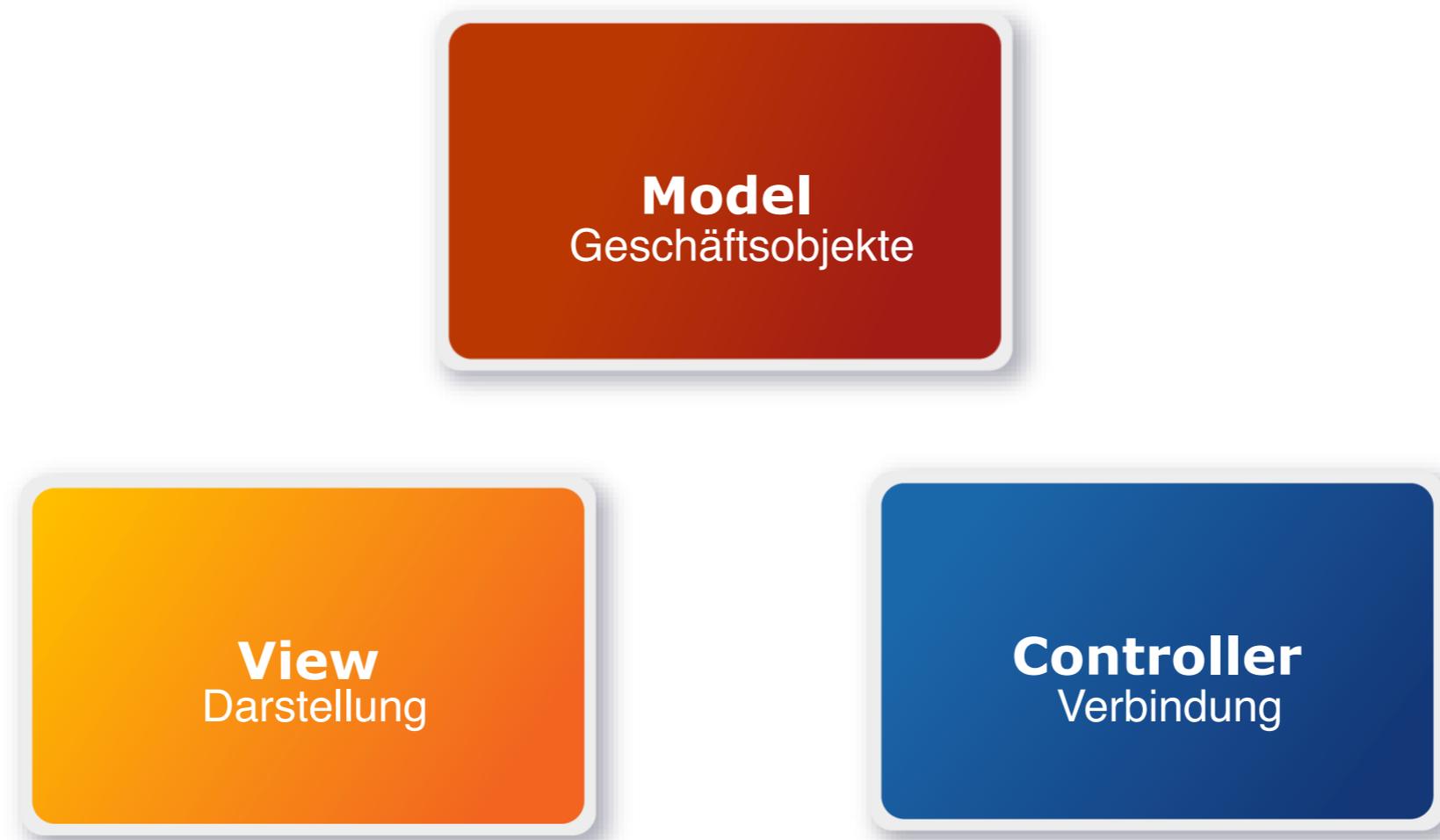
AngularJS-Applikation

AngularJS-Application

- Die Direktive `ng-app="{appname}"` definiert eine AngularJS-Application.
- Durch die Bindung an ein HTML-Element wird der Gültigkeitsbereich definiert.
- `angular.module('app', [{Dependency Injection}]);`
definiert das zugehörige Module.
- Über Dependency Injection werden Abhängigkeiten zu anderen Modulen definiert.

Model-View-Controller

- Trennung von Geschäftsobjekten und Darstellung



```

<body ng-app="app">
<div class="container" ng-controller="GuessTheNumberController">
  <h2>Guess the Number !</h2>
  <p class="well lead">Guess the computer generated random number between 1 and 1000.</p>
  <label>Your Guess: </label><input type="number" ng-model="guess"/>
  <button ng-click="verifyGuess()" class="btn btn-primary btn-sm">Verify</button>
  <button ng-click="initializeGame()" class="btn btn-warning btn-sm">Restart</button>
  <p>
    <p ng-show="deviation<0" class="alert alert-warning">Your guess is higher.</p>
    <p ng-show="deviation>0" class="alert alert-warning">Your guess is lower.</p>
    <p ng-show="deviation===0" class="alert alert-success">Yes! That's it.</p>
  </p>
  <p class="text-info">No of guesses : <span class="badge">{{noOfTries}}</span><p>
</div>
<script src="../bower_components/angular/angular.js"></script>
<script type="text/javascript">
  angular.module('app', [])
  .controller('GuessTheNumberController', GuessTheNumberController);
  function GuessTheNumberController($scope) {
    $scope.verifyGuess = function () {
      $scope.deviation = $scope.original - $scope.guess;
      $scope.noOfTries = $scope.noOfTries + 1;
    }
    $scope.initializeGame=function() {
      $scope.noOfTries = 0;
      $scope.original = Math.floor(Math.random() * 1000) + 1);
      $scope.guess = null;
      $scope.deviation = null;
    }
    $scope.initializeGame();
  }
</script>
</body>

```

Controller

Controller

Controller

- Die Direktive `ng-controller="{ControllerName}"` definiert den Controller.
- Durch die Bindung an ein HTML-Element wird der Gültigkeitsbereich definiert.
- ```
angular.module('app', [{Dependency
Injection}]).controller('GuessTheNumberController', function($scope) {});
```

  
registriert den Controller im Module.

```

<body ng-app="app">
<div class="container" ng-controller="GuessTheNumberController">
 <h2>Guess the Number !</h2>
 <p class="well lead">Guess the computer generated random number between 1 and 1000.</p>
 <label>Your Guess: </label><input type="number" ng-model="guess"/>
 <button ng-click="verifyGuess()" class="btn btn-primary">Verify</button>
 <button ng-click="initializeGame()" class="btn btn-primary">Restart</button>
 <p>
 <p ng-show="deviation<0" class="alert alert-warning">Your guess is higher.</p>
 <p ng-show="deviation>0" class="alert alert-warning">Your guess is lower.</p>
 <p ng-show="deviation===0" class="alert alert-success">Yes! That's it.</p>
 </p>
 <p class="text-info">No of guesses : {{noOfTries}}</p>
</div>
<script src="../bower_components/angular/angular.js"></script>
<script type="text/javascript">
 angular.module('app', [])
 .controller('GuessTheNumberController', GuessTheNumberController);
 function GuessTheNumberController($scope) {
 $scope.verifyGuess = function () {
 $scope.deviation = $scope.original - $scope.guess;
 $scope.noOfTries = $scope.noOfTries + 1;
 }
 $scope.initializeGame=function() {
 $scope.noOfTries = 0;
 $scope.original = Math.floor(Math.random() * 1000) + 1);
 $scope.guess = null;
 $scope.deviation = null;
 }
 $scope.initializeGame();
 }
</script>
</body>

```

Model

Model

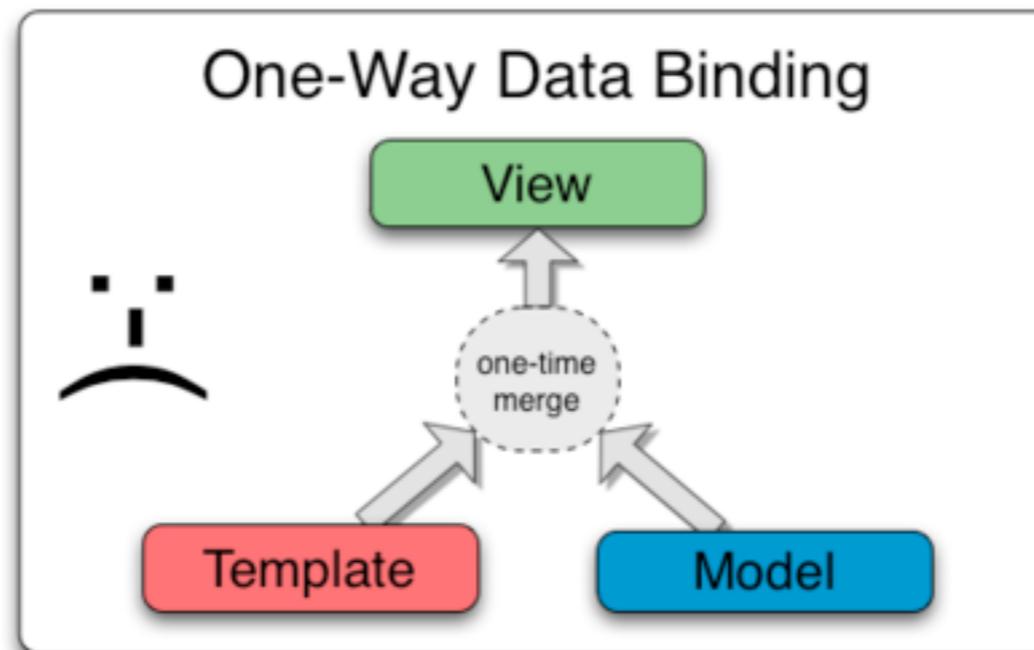
Model

## Model

- Das Model wird über die `$scope`-Variable des Controllers zugänglich gemacht.
- Input-Elemente werden mit der Direktive `ng-model="{Variablenname}"` an das Model gebunden.
- Angular verwendet ein Two-Way Datebinding.

## One-Way Data Binding

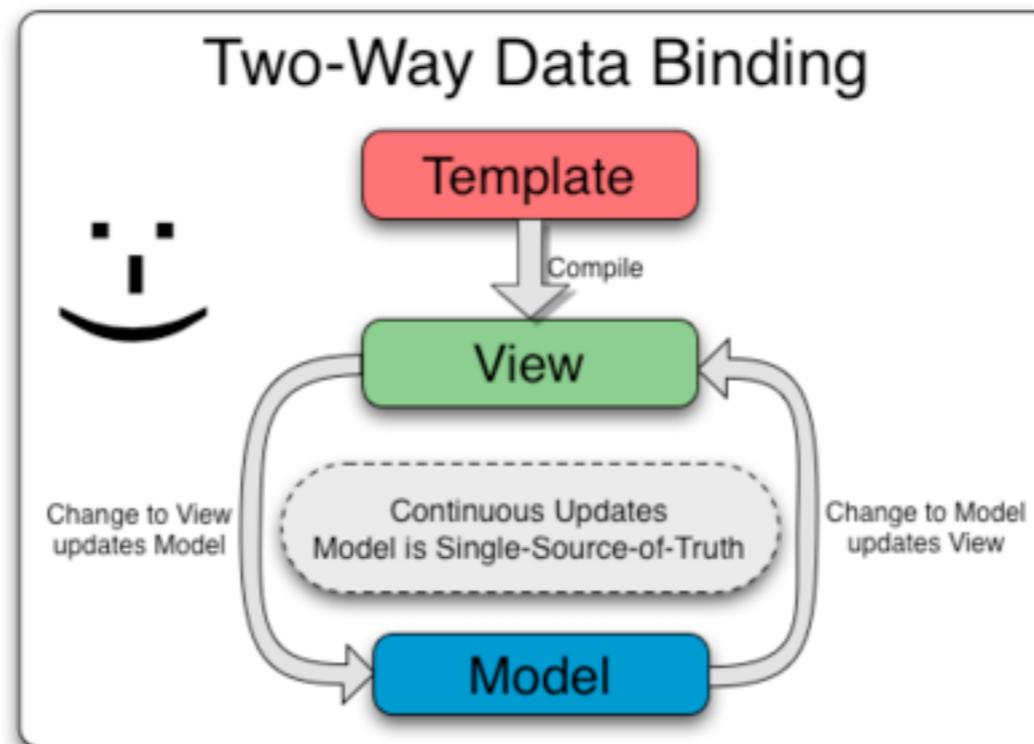
- Andere JavaScript Frameworks bringen zum Zeitpunkt des Renderings die Daten aus dem Model mit einer Vorlage zusammen.
- Änderungen aus der View (HTML-Seite) haben keine direkte Auswirkung auf das Model.



Grafik dem AngularJS Tutorial entnommen: <https://docs.angularjs.org/guide/databinding>

## Two-Way Data Binding

- In Angular wirken sich Änderungen in HTML-Input-Elementen unmittelbar auf das Model und vice versa aus.
- Single-Source-of-Truth!



Grafik dem AngularJS Tutorial entnommen: <https://docs.angularjs.org/guide/databinding>

## Angular Expression

- Angular Expression sind kleine Code Snippets.
- Meistens werden sie in doppelte geschweifte Klammern `<span class="badge">{{noOfTries}}</span>` gepackt.
- Angular Expressions unterscheiden sich von JavaScript Expressions durch
  - Kontext bezieht sich auf den aktuellen Scope anstelle des Window Objektes
  - Fehlertoleranter
  - Keine Schleifen, If-Abfragen und ähnliches
  - Nicht anfällig für Cross-Site Scripting (xss)
  - Angular Filter können benutzt werden

## One-Time Binding

- Die bidirektionale Datenbindung bedeutet einen gewissen Overhead.
- Wird dieser permanente Abgleich nicht benötigt, kann eine so genannte One-Time Binding eingesetzt werden.
- In der Expression wird durch vorgestellte zwei Doppelpunkte signalisiert, dass der Wert nur einmal beim Laden der Seite ausgewertet werden soll.

```
<h1>{{ ::model.title }}</h1>
```

- Vergleichbar in XPages mit
  - **Compute Dynamically**  
value="#{javascript:database.getTitle()}"
  - **Compute on Page Load**  
value="\$ {javascript:database.getTitle()}"

## Single Page Application - SPA

- Bei SPA wird initial nur minimaler HTML Code geladen.
- Jegliche weitere Inhalte werden entweder
  - per JavaScript berechnet
  - per AJAX nachgeladen.
- In Abhängigkeit vom Anwendungsfluss werden Teile ausgetauscht bzw. nachgeladen.
- Herausforderungen sind
  - Lesezeichen setzen
  - Back-Button
  - Navigation zwischen verschiedenen „Seiten“

## Routen - ngRoute

- In Angular wird diese Aufgabe vom \$route Service übernommen.
- Der zugehörige Code muss als separate JS-Datei „angular-route.js“ geladen werden.
- Als Dependency muss „ngRoute“ definiert werden.
- Die Wege zwischen den Teilseiten werden als sogenannte Routen definiert.

## Form Management

- Angular bietet eine Reihe von Hilfsmitteln, um den Umgang mit HTML Forms zu vereinfachen.

## Server Kommunikation

- Für die Kommunikation mit dem Server gibt es zwei primäre Services in Angular
- `$http` - AJAX Basis-Operationen
- `$resource` - Höhere Abstraktionsebene für die Verwendung von REST
  - `$resource` basiert auf `$http`

## Fragen?

jetzt stellen – oder später:

 [bhort@assono.de](mailto:bhort@assono.de)

 <http://www.assono.de/blog>

 040/73 44 28-315



Folien unter

[www.assono.de/blog/d6plinks/EntwicklerCamp-2016-AngularJS](http://www.assono.de/blog/d6plinks/EntwicklerCamp-2016-AngularJS)