

Rich Internet Applications mit HTML5 und Javascript



Thorsten Wirth

Agenda

- ▶ Was ist AngularJS
- ▶ Grundlagen
 - ▶ MVC-Muster
 - ▶ MVVM-Muster
 - ▶ Datenbindung und Scopes
 - ▶ Dependency Injection (DI)
 - ▶ Testbarkeit

Agenda

- ▶ Bestandteile von AngularJS
 - ▶ Module
 - ▶ Controller
 - ▶ Models
 - ▶ Routen
 - ▶ Templates und Expressions
 - ▶ Filter
 - ▶ Services
 - ▶ Direktiven

Agenda

- ▶ Eine kleine „todoApp“
- ▶ Fazit

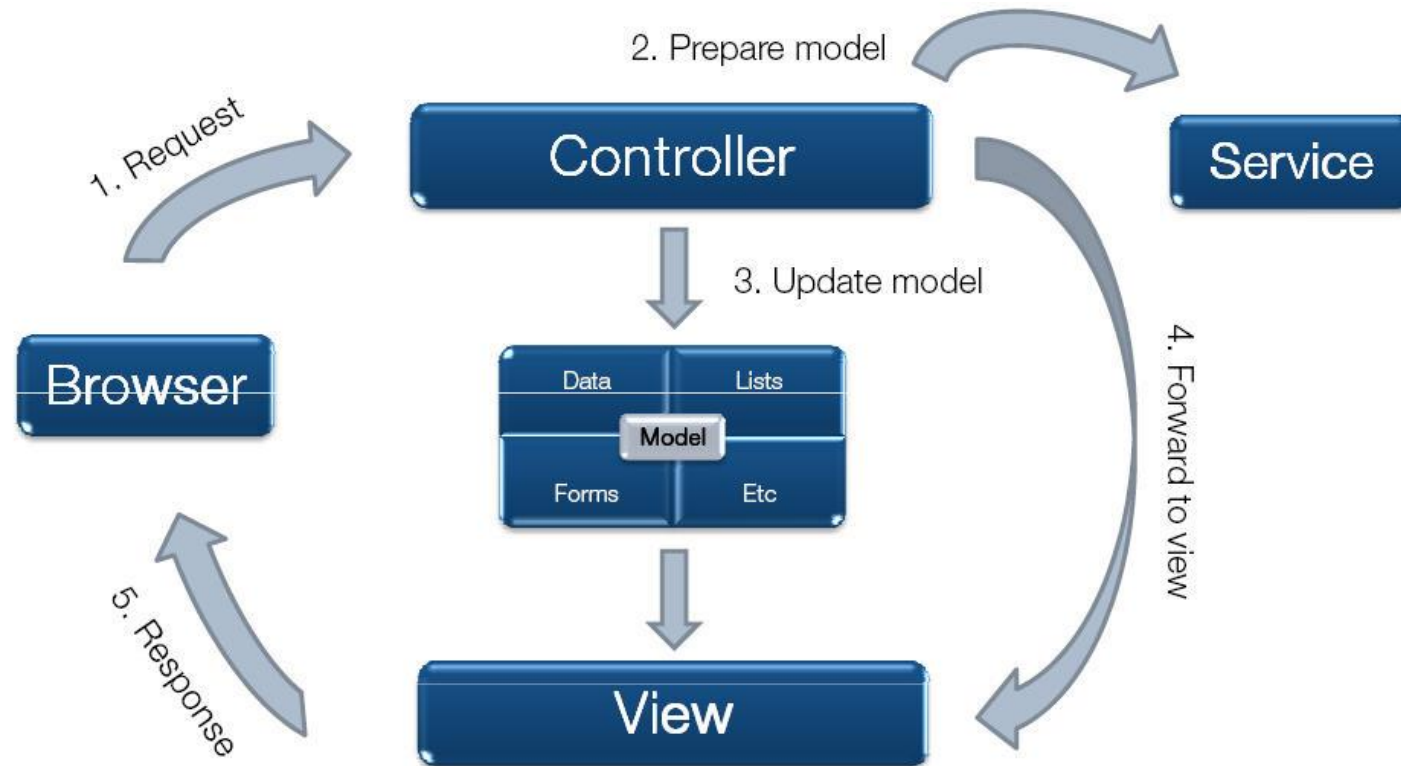
Was ist AngularJS

- ▶ Open-Source-Framework
- ▶ Komplet in JavaScript
- ▶ Entwickelt von Google Inc
- ▶ Erscheinungsjahr 2009
- ▶ MIT-Lizenz

Grundlagen - MVC

- ▶ MVC erstmals in den 80er Jahren durch Trygve Reenskaug beschrieben
- ▶ Zuerst für die Programmiersprache Smalltalk
- ▶ Trennung der Geschäftslogik (Controller), Datenhaltung (Model) und der Präsentation (View)

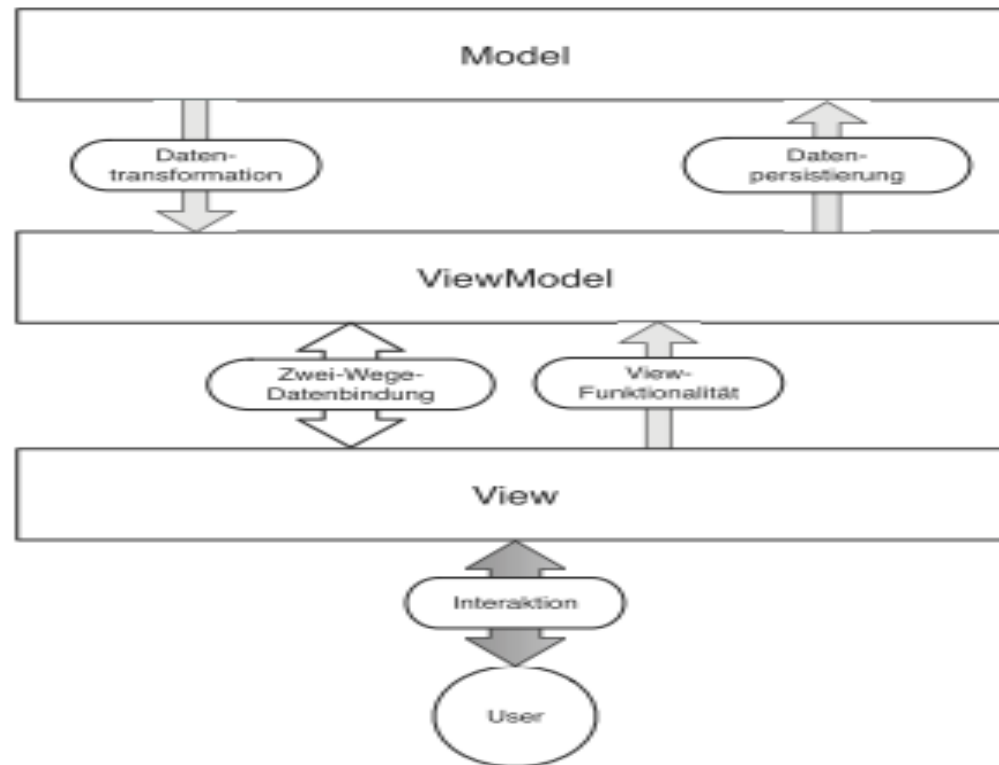
Grundlagen - MVC



Grundlagen - MVVM

- ▶ Nutzt die Trennung wie im MVC
- ▶ ViewModel ist eine Art Proxyschicht die zwischen View und Model vermittelt
- ▶ Beinhaltet die UI-Logik
- ▶ Zwei-Wege-Datenbindung
- ▶ Wenig „Boilerplate-Code“

Grundlagen - MVVM



Grundlagen - Datenbindung und Scopes



- ▶ Datenbindung nur bei Variablen und Funktionen, die in einem Scope definiert wurden.
- ▶ Scope bezieht sich auf einen bestimmten Abschnitt im DOM
- ▶ Änderung der Daten über die „Dirty Checking“ Methode

Grundlagen – Dependency Injection

- ▶ Auflösung der Abhängigkeiten wird an eine zentrale Stelle übergeben
- ▶ Erhöht die Austauschbarkeit und Testbarkeit der Anwendung
- ▶ Gehört zu den Leitkonzepten in AngularJS
- ▶ Eines der Alleinstellungsmerkmale unter den JavaScript-Frameworks

Grundlagen - Testbarkeit

- ▶ Ein Konzept bei der Entwicklung von AngularJS, lag bei der Testbarkeit der Anwendungen
- ▶ Von Haus aus Konzepte für Unit-Tests und E2E-Tests an Bord.
- ▶ E2E-Tests ermöglichen das Testen des Verhaltens einer ganzen Anwendung

Bestandteile von AngularJS

Module



- ▶ Module bilden einen Rahmen, in dem zusammenhängende Controller, Direktiven, Services und Filter gekapselt werden.
- ▶ Auch bei den Modulen wird DI angewendet, so dass in einem Modul weitere Module eingebunden werden können.

Bestandteile von AngularJS

Controller

- ▶ Regeln bei Controllern:
 - ▶ So schmal wie möglich halten
 - ▶ Komplexe Logik in Services auslagern und via DI in Controller einbinden
 - ▶ Code sollte aussagekräftig sein
 - ▶ Absolutes No-Go:
 - ▶ DOM-Manipulation
- ▶ Eigentlich nur ein Konstruktor für Scopes
- ▶ Scopes:
 - ▶ Stellen Daten und Logik für eine bestimmte Ansicht bereit

Bestandteile von AngularJS Models



- ▶ Sind gewöhnliche JavaScript-Datentypen.
- ▶ Keine besonderen Klassen von denen geerbt werden müsste.
- ▶ Models werden in einem Scope definiert, um die Zwei-Wege-Datenbindung herzustellen

Bestandteile von AngularJS Routen

```
angular.module('myApp').config(function ($routeProvider) {
  $routeProvider
    .when('/', {
      templateUrl: 'templates/mainTemplate.html',
      controller: 'MainCtrl'
    })
    .when('/user/:userId', {
      templateUrl: 'templates/userDetailsTemplate.html',
      controller: 'UserDetailsCtrl'
    })
    .when('/user', {
      templateUrl: 'templates/userOverviewTemplate.html',
      controller: 'UserOverviewCtrl'
    })
    .otherwise({
      redirectTo: '/'
    });
});
```


Bestandteile von AngularJS Templates und Expressions



- ▶ Mit den Templates wird die Website zusammen gesetzt.
- ▶ Expressions sind für die Zwei-Wege-Datenbindung unerlässlich.
- ▶ Expressions werden verwendet um in einem Template auf Daten, die in einem Scope definiert wurden, zuzugreifen.
- ▶ Sie werden in zwei geschweifte Klammern geschrieben.
- ▶ Ändern sich die Daten im Scope, wird auch die Expression automatisch neu evaluiert.

Bestandteile von AngularJS

Filter

- ▶ Werden eingesetzt, um Expression vor der Ausgabe zu transformieren.
- ▶ 2 Arten von Filtern:
 - ▶ Formatierungsfiler:
 - ▶ Formatieren die Expressions vor ihrer Ausgabe
 - ▶ Collection-Filter:
 - ▶ Finden ihre Anwendung in Verbindung mit der Direktiven „ng-repeat“.
- ▶ Filter werden in AngularJS mit einem Pipe eingeleitet.

Bestandteile von AngularJS Services

- ▶ In ihnen wird komplexere Logik implementiert, die in den Controllern keinen Platz findet,
- ▶ Werden häufig auch als DAO eingesetzt.
- ▶ Werden nur einmal instantiiert (Singleton-Pattern) und eigenen sich dadurch Daten und Logik in mehreren Controllern zu verwenden.
- ▶ Es gibt 5 Möglichkeiten Services zu definieren. Die häufigste verwendete Funktion ist die „factory()“.

Bestandteile von AngularJS

Direktiven



- ▶ In keinem anderen Framework zu finden.
- ▶ Beliebige Erweiterung von HTML um weitere Tags oder Attribute.
- ▶ Beispiel einer Direktive in AngularJS:
 - ▶ ng-repeat
- ▶ Bei Erstellung eigener Direktiven Namenskonvention beachten (Beginn mit Kleinbuchstabe danach CamelCase-Notation)
- ▶ Verwendung im Template dann mit „-“, „:“ und „_“

Bestandteile von AngularJS

Direktiven

- ▶ 2 Möglichkeiten einer Direktiven-Definition
 - ▶ Link-Funktion gibt eine Funktion zurück
 - ▶ DDO gibt ein Direktiven-Definitions-Objekt zurück
- ▶ DDO wesentlich detaillierter und für Anfänger sehr komplex.

Fazit

- ▶ 1. Phase (Einarbeitung) = steile Lernkurve = sehr hohe Motivation
- ▶ Setzt Kenntnisse in JavaScript voraus.
- ▶ Zunehmend schwieriger in das Thema vorzudringen, da komplexere JavaScript Funktionen zum Einsatz kamen.
- ▶ Nachvollziehen des Zusammenwirkens der Bestandteile forderte einiges an Zeit.

Fazit

- ▶ Positiv:
 - ▶ Zwei-Wege-Datenbindung
 - ▶ Schlanker Code und automatische Aktualisierung
 - ▶ Bausteine lassen sich in separaten Dateien einfach auslagern -> Übersichtlichkeit ↑
 - ▶ Wirkt sehr ausgereift und große Community
- ▶ Negativ:
 - ▶ Nur für Single Page Anwendungen mit viel Interaktion mit dem User (sonst Aufwand zu groß)