1.– 4. September 2014
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Die Grundlagen

Philipp Burgmer

theCodeCampus / Weigle Wilczek GmbH

ABOUT ME

Philipp Burgmer

- Software Engineer / Consultant / Trainer
- Focus: Frontend, Web Technologies

- WeigleWilczek GmbH
- burgmer@w11k.com

ABOUT US

WeigleWilczek / W11k

- Software Design, Development & Maintenance
- Consulting, Trainings & Project Kickoff

- Web Applications with AngularJS
- Native Rich Clients with Eclipse RCP

WEB APPS UNTIL NOW

- JSF
  - UI on Server
  - A lot HTTP Requests Just to Update UI
  - Hard to Use JS Libs / Scatters UI Logic
- GWT
  - UI in Java / XML
  - Hard to Use JS Libs / Scatters UI Logic
  - "Java World" Instead of "Web World"
- Flex
  - Clean Separation of Front- and Backend
  - Based on Flash, Adobe Discontinues Developement
  - MXML and ActionScript Instead of HTML and JavaScript

WEB APPS FROM NOW ON

- Frontend Runs Completely in the Browser
- Stateful UI, Stateless Server
- Server Delivers Static Resources
- Server Delivers Dynamic Data
- HTML, CSS and JavaScript as UI Toolkit

WHAT IS ANGULARJS?

> *HTML Enhanced for Web Apps*
>
> **angularjs.com**

- Client / Browser JS Framework
- Rich Browser Applications
- Brings Core Frontend Concepts and Features to the Browser
- Extends HTML Instead of Abstracting or Wrapping It

- angularjs.org
- Current Versions: 1.2.23 and 1.3.0-beta.19
- License: MIT

CORE CONCEPTS

- Model View Controller Pattern
- Two Way Data-Binding
- Directives
- Modules
- Dependency Injection
- Services
- Filter

Goals

- Separation of Concerns
- Make It Easier to Write Clean Code
- Make It Easier to Write Testable Code
- Offer Concepts and Be Open for Extensions

DEMO

- Two Way Data-Binding [JS Bin | localhost]
- Add Logic with a Controller [JS Bin | localhost]
- Format Values with Filters [JS Bin | localhost]

DEPENDENCY INJECTION

JAVA WITH GOOGLE GUICE

```
1  public interface Service {}
2
3  @Singleton
4  public class ServiceImpl implements Service {
5    @Inject
6    public ServiceImpl(final OtherService otherService) { }
7  }
```

JAVA WITH GOOGLE GUICE

```
public class MyModule extends AbstractModule {
  protected void configure() {
    // bind with interface
    bind(Service.class).to(ServiceImpl.class);
    // bind with scope
    bind(OtherService.class).in(Singleton.class);
    // bind with alias
    bindConstant().annotatedWith(Names.named("port")).to(8080);
  }
}
```

JAVA WITH GOOGLE GUICE

```
1 // manually or by configured framework
2 final Injector injector = Guice.createInjector(new MyModule());
3 final Service service = injector.getInstance(Service.class);
```

JAVASCRIPT WITH ANGULARJS

```
angular.module('myModule', ['moduleOfOtherLibrary'])
// no scopes, services are singletons by definition
.service('usefulService', function($window) {
  function somethingPrivate() { }

  return function() {
    somethingPrivate();
    $window.close();
  }
};
```

JAVASCRIPT WITH ANGULARJS

```
angular.module('myModule', ['moduleOfOtherLibrary'])
// no scopes, services are singletons by definition
.service('usefulService', function(a) {
  function somethingPrivate() { }

  return function() {
    somethingPrivate();
    a.close();
  }
};
```

JAVASCRIPT WITH ANGULARJS

```
angular.module('myModule', ['moduleOfOtherLibrary'])
// no scopes, services are singletons by definition
.service('usefulService', ['$window', function(a) {
  function somethingPrivate() { }

  return function() {
    somethingPrivate();
    a.close();
  }]
};
```

JAVASCRIPT WITH ANGULARJS

```
var service = function(a) {
  return function() {
    a.close();
  }
}
service.$inject = ['$window'];

angular.module('myModule', ['moduleOfOtherLibrary'])
.service('usefulService', service);
```

JAVASCRIPT WITH ANGULARJS

ADDITIONAL PARAMETERS AND OVERRIDDEN DI VALUES

```
// get the injector via static call
var $injector = angular.injector();
// or via injection
function($injector) { }
```

```
var functionA = function(serviceA) { };
$inject.invoke(functionA);

var functionB = function(serviceA, nonDIValue) { };
var locals = { nonDIValue: 1 };
$inject.invoke(functionB, locals);
```

DIRECTIVES

WHAT IS IT?

- Extends HTML
- Elements, Attributes, CSS classes
- Encapsulates DOM Manipulations
- Found and Proceeded by AngularJS
- Similar to Web-Components But Usable Today

USING DIRECTIVES

```
<!-- 'my-directive' as element -->
<my-directive></my-directive>

<!-- 'my-directive' as attribute -->
<span my-directive="..."></span>

<!-- 'my-directive' as css class -->
<span class="my-directive"></span>
```

DEFINING DIRECTIVES

```
angular.module('app', [])
.directive('myDirective', function() {
  return {
    restrict: 'E',
    replace: true,
    transclude: true,
    template: '<span ng-transclude=""></span>',
    link: function(scope, element, attrs) {
      // do some DOM manipulation
    }
  };
});
```

DEMO

- Blink on ~~Steroids~~ Speed [JS Bin | localhost]
- New Tags with Directives [JS Bin | localhost]
- Fancy Stuff: Bouncy Balls [JS Bin | localhost]

BUILD-IN FEATURES & COMPONENTS

FEATURES

- Async Programming with Promises
- Forms & Validation

- Templating
- Routing
- Animations
- REST Communication
- Mocking
- ...

COMPONENTS

Directives

- ng-click
- ng-class
- ng-show
- ng-include
- ng-view
- ng-change
- ng-model
- ng-repeat
- ...

Services

- $http
- $location
- $log
- $q
- $resource
- $route
- $timeout
- $window
- ...

Filter

- currency
- date
- filter
- json
- limitTo
- lowercase
- number
- orderBy
- ...

VIEWS & ROUTES

- Deep linking
- Partial Views / Templating
- Optional JS File and Module `ngRoute`

```
angular.module('myApp').config(function ($routeProvider) {
  $routeProvider.when('/home', {
    templateUrl : "route/home.html",
    controller : 'HomeCtrl'
  })
  .otherwise({
    redirectTo: '/home'
  });
});
```

```
<div class="content">
  <div ng-view></div>
</div>
```

DEMO

- Small CRUD App [JS Bin | localhost]
- This Presentation

ANIMATIONS

- New in Version 1.2
- Optional JS File and Module `ngAnimate`
- Plain CSS Animations and Transitions
- CSS Classes for 'enter', 'move' and 'leave'
- Custom JS Functions Possible
- Directives Must Support Animation via `$animate`

- ng-repeat
- ng-view
- ng-include
- ng-show
- ng-hide

DEMO

- ng-repeat [JS Bin | localhost]
- ng-view [JS Bin | localhost]

ECO SYSTEM

TOOLS

Task Runner / Build System

Bower

Dependency Management for
3rd party libs

Bootstrap

Front-End Framework
Styling & Layout

CSS Extension Language

Version Control System

Runtime for Dev Tools /
Backend Environment

FABS

FABULOUS ANGULARJS BUILD SYSTEM

- Helps to Structure Code (Feature-Oriented-Structure)
- Dev-Mode with Server, Proxy and LiveReload
- SASS and LESS Support
- Spec and End-2-End Test
- Mock Data for Tests and Developing
- Project- and Per-Developer Configuration
- Building Distribution
  - Annotating AngularJS Dependencies (Transform to Array-Notation)
  - Code Minimization
  - Running End-2-End Tests Against Build Application

- github.com/w11k/fabs
- Yeoman Generator: `generator-fabs`

USEFUL MODULES & LIBRARIES

AngularJS

- ngmodules.org
- Our Modules: github.com/w11k

- Angular-UI Bootstrap and/or AngularStrap
- Angular-UI Router as Alternative to ngRoute
- Angular-Translate for l10n and i18n
- Angular-Growl for Notifications
- Angular Bind-Once for High-Performance-Bindings

JavaScript

- Lodash / Underscore for Functional Programming
- MomentJS for Date Parsing and Calculations
- D3 and/or Highcharts for Data-Visualization and Charts

CONCLUSION

- Clean Separation of Frontend and Backend
- DI, MVC and DataBinding in the Browser
- Seamless Integration with other Frameworks

- Easy to Learn (with Some Harder Parts Later on)
- Documentation with a Lot of Runnable Examples
- Large Community and Fast Growing Eco System
- Powered and Used by Google

TRY IT!

Philipp Burgmer
burgmer@w11k.com

www.w11k.com
www.thecodecampus.de