

Karlsruher Entwicklertag 2013

Web Applications with



About me

Philipp Burgmer

Senior Software Engineer / Consultant

WeigleWilczek GmbH

burgmer@weiglewilczek.com

Main Focus: Frontend, Web Technologies

Web Apps until now

GWT

UI in Java / XML

hard to use JS libs / scatters ui logic

"Java World" instead of "Web World"

JSF

UI on Server

a lot HTTP requests just to update UI

hard to use JS libs / scatters ui logic

Flex

based on Flash

Adobe discontinues development

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

lets you extend html to fit your needs

Core Concepts

Model View Controller Pattern

Two Way Data-Binding

Dependency Injection

Modules

Services

Directives

Filter

Separation of Concerns

Testable Code

Demo

~~Two Way Data-Binding~~

~~Add Logic with a Controller~~

~~Format Values with Filters~~

~~Adjust markup with Filters~~

Directives

extend HTML

Tags, Attributes, CSS classes

encapsulate DOM manipulations

preceded by AngularJS compiler

Demo

~~Blink on Steroids Speed~~

~~New Tags with Directives~~

Views & Routes

Deep linking

Partial Views / Templating

```
function $routeProvider(){  
    this.when = function(path, route) { };  
    this.otherwise = function(params) { };  
    this.$get = function($rootScope, $location, $routeParams, $q, $injector, $http, $templateCache) {  
        ...  
        return $route;  
    };  
}
```

Demo

Small crud app -> with own URL bar: local

Fancy stuff: bouncy balls

This Presentation

Animations

new in Version 1.2

easy to use

plain CSS Animations and Transitions

CSS class names for 'enter', 'move' and 'leave'

custom JS functions possible

open for animation libs

directives must support ng-animate

ng-repeat

ng-view

ng-include

ng-show

ng-hide

Demo

ng-repeat

ng-view

Built-in Features

Extensibility

Templating

Localization

Validation

REST support

Embeddable

Testable

Built-in Features

Directives

ng-click
ng-class
ng-show / ng-hide
ng-include
ng-view
ng-pluralize
ng-repeat
ng-submit
...

Filter

currency
date
filter
json
limitTo
lowercase
number
orderBy
uppercase

Services

http
location
log
q
resource
route
timeout
window
...

Conclusion

Clean separation of Frontend and Backend

Features like DI, MVC and DataBinding in the browser

Seamless integration with other frameworks

Lets you use all the cool new Web / JS tools

Easy to learn

Documentation with a lot of runnable examples

Large community and fast growing eco system

powered and used by Google

Try it!

Philipp Burgmer
burgmer@weiglewilczek.com

Dependency Injection

Java with Google Guice

```
// no dependency management
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);
    }
}
```

Dependency Injection

Java with Google Guice

```
@Singleton  
public class ServiceImpl {  
    @Inject  
    public ServiceImpl(final OtherService otherService) { }  
}
```

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

Dependency Injection

JavaScript with AngularJS

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

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

Dependency Injection

JavaScript with AngularJS

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

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

Dependency Injection

JavaScript with AngularJS

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

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

Dependency Injection

JavaScript with AngularJS

```
var service = function(a) {  
    return function() {  
        a.close();  
    }  
}  
  
service.$inject = ['$window'];  
  
angular.module('myModule', ['moduleOfOtherLibrary'])  
    .service('usefulService', service)  
    .constant('port', 80)
```


Dependency Injection

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);
```

Philipp Burgmer
burgmer@weiglewilczek.com