

# Angular

## Angular (Zusammenfassung September 2015)

### Promises

In angular \$q-service: Q: Asynchronous promises in JavaScript: <https://github.com/kriskowal/q>

From the perspective of dealing with error handling, deferred and promise APIs are to asynchronous programming what `try`, `catch` and `throw` keywords are to synchronous programming.

### Providers

Providers are objects that provide (create) instances of services and expose configuration APIs that can be used to control the creation and runtime behavior of a service. In case of the `$route` service, the `$routeProvider` exposes APIs that allow you to define routes for your application.

Providers can only be injected into `config` functions. Thus you could not inject `$routeProvider` into `PhoneListCtrl`.

<https://github.com/angular/angular.js/wiki/Understanding-Dependency-Injection>

### `$scope.$apply`

<http://www.jeffryhouser.com/index.cfm/2014/6/2/How-do-I-run-code-when-a-variable-changes-with-AngularJS>

### Microsoft.WindowsAzure.MobileServices, MobileServiceClient

reading/writing into table of MobileService, also demonstrates `async`, `await` (see separate note)

### Javascript IntelliSense in Visual Studio (doesn't work until now)

<http://weblogs.asp.net/scottgu/vs-2008-javascript-intellisense>

<http://blogs.msdn.com/b/webdev/archive/2007/11/06/jscript-intellisense-a-reference-for-the-reference-tag.aspx>

## Angular Style Guide

<https://github.com/johnpapa/angular-styleguide#controlleras-with-vm>

## Angular Formly Example

<http://angular-formly.com/#/example/intro/introduction>

<http://docs.angular-formly.com/>

# Angular (Zusammenfassung von Scott Allens presentation)

Scott Allen, odetocode

noch machen: An AngularJS Playbook/Forms Playbook

Verwende Plunker zum üben

Javascript pattern:

abstraction: verwende Funktionen um Implementierung zu verstecken

revealing pattern: eine funktion A liefert ein javascript-Objekt, das als Felder Funktionen drin hat. Funktion A hat lokale Funktionen, die gegen aussen nicht sichtbar sind

immediatly invoked function IIF:

Eine Funktion wird definiert und danach grad auch noch ausgeführt.

Vorteil: Es können lokale Funktionen definiert werden und der global scope wird nicht unnötig mit Symbolen verschmutzt.

`( function() {} ) () );`

ng=angular

directives=used as html-attributes, e.g. ng-app, ng-controller, ng-src usw.

angular arbeitet nur auf dem html-element, wo html-attribut ng-app dran ist

Test um zu sehen, ob ng richtig installiert wurde:

```
{{2+2}}
```

ng-controller directive:

Enthält Name der Javascript-Funktion in global scope. (Nicht mehr unterstützt mit aktuellem angular)

Aktuell: Name des controllers im modul das bei ng-app definiert wurde.)

angular ist das einzige Symbol von angular im global scope

```
var module1=
```

```
angular.module('module1',[]);
```

// [] definiert Abhängigkeit zu anderen Modulen. Wenn [] nicht vorhanden, wird versucht, ein bestehendes Modul zu finden und es wird null geliefert falls noch keinez vorhanden.

```
module1.controller('controller1', controller1)
```

ng-model: bindet input-element zu model-field

ng-show/ng-hide: ein div wird sichtbar, unsichtbar

ng-repeat: iteriert über array

ng-click: ruft function in model auf

ng-include: inkludiert eine andere html-Seite

built-in services of angular:

\$http, \$interval, \$location

Vorteile von services in angular:

es gibt nur eine einzige Instanz, die aus allen Controllern verwendet werden kann. Redundanten Code aus Controller nach Service verschieben.

eigener Service in separatem js-file:

```
modul.factory('service1', function)
```

routing: Abhängig von der URL im Browser, wird ein anderer Controller und ein anderes HTML-File verwendet.

Der Router wird im Modul konfiguriert, und definiert pro URL je einen Controller und ein HTML-File.

Innerhalb des Controllers kann zu einer anderen URL navigiert werden.

react.js: kann als View bei Angular verwendet werden

Backend as Service:

[https://www.backand.com/apps/#/sign\\_up](https://www.backand.com/apps/#/sign_up)

# Angular.js (May 2015)

**Tools (last updated: 2015-05-14)**

- \* tutorial: <https://github.com/angular/angular-phonecat> (mit **git clone** holen)
- \* NetBeans (New Project, HTML5, HTML5 Applications with Existing Sources)
- \* git command line (git clone, git add, git commit, aber wie geht sync ?)
- \* GitHub for mac (besser als comandline)
- \* npm
- \* <https://www.npmjs.com/package/http-server>
- \* <https://cloudant.com/> (later for persisting data)

## AngularJS Do's and Dont's

<http://blog.nebithi.com/angularjs-dos-and-donts/>

## Angular documentation

<https://docs.angularjs.org/guide/directive>

<http://angular-ui.github.io/bootstrap/#/top>

<https://www.jqwidgets.com/jquery-widgets-documentation/documentation/angularjs/angularjs.htm>