

# Kurzreferenz („Cheat Sheet“) AngularJS

Autoren: FH-Prof. Manfred Steyer (M.Steyer@IT-Visions.at), Dr. Holger Schwichtenberg (H.Schwichtenberg@IT-Visions.de)

## Scripteinbindung

```
<script src="/xy/angular.js"></script>
```

## Controller

### Controller erzeugen

```
var app = angular.module("Flug", []);
app.controller("FlugBuchenCtrl", function ($scope, $http, $q) {
    $scope.vm = new FlugBuchenVM($scope, $http, $q);
});
```

### Angabe der Abhängigkeiten bei Einsatz von Minification

```
app.controller("FlugBuchenCtrl", ["$scope", "$http", "$q",
function ($scope, $http, $q) {
    $scope.vm = new FlugBuchenVM($scope, $http, $q);
}]);
```

## Direktiven

### Direktiven, um auf Controller zu verweisen

```
<div ng-app="flug">
    <div ng-controller="FlugBuchenCtrl">
        [...]
    </div>
</div>
```

### Bedingtes Anzeigen

```
<table ng-show="vm.fluege.length > 0">
</table>
```

### Bereiche für Auflistungen wiederholen

```
<tr ng-repeat="f in vm.fluege track by $index">
    <td>{{f.Datum}}</td>
    <td>{{f.Abflugort}}</td>
    <td>{{f.Zielort}}</td>
</tr>
```

### Bedingtes Formatieren

```
<div ng-class="{ selected: f.Id == vm.selectedFlug.Id }" />
```

## Methodenaufrufe

### Click-Handler ohne Parameter

```
<input type="button" value="Suchen"
      ng-click="vm.loadFluege()" />
```

## HTTP

### Ressourcen via GET anfordern

```
params = {
    flugNummer: that.flugNummerFilter
};
this.http.get("/api/flug", { params: params })
    .then(function (result) {
        var body = result.data;
    }).catch(function (reason) {
        that.message = "Fehler: " + reason;
    });
}
```

### Ressourcen via POST senden

```
var buchung = {
    FlugID: that.selectedFlug.Id,
    PassagierID: that.selectedPassagier.Id
};
that.http.post("/api/buchung", buchung)
    .then(function (result) {
        that.message = "Gebucht!";
    }).catch(function (reason) {
        that.message = "Fehler: " + reason;
    });
```

### Standard-Header setzen

```
$http.defaults.headers.common.Authentication = "..."
```

## Services

### Factory für Service definieren

```
var app = angular.module("Flug", []);
[...]
app.factory("flugService", function ($http, $q) {
    return { ... }
});
```

### Minification-sichere Factory

```
app.factory("flugService", ["$http", "$q", function ($http, $q)
{ return { ... } }]);
```

## Filter

### Ausgewählte Beispiele vordefinierter Filter

```
<td>{{p.Geburtsdatum | date:'shortDate' }}</td>
<td>{{p.Preis | number:2}}</td>
<td>{{ eigenschaft | json }}</td>
<tr ng-repeat="p in vm.passagiere | filter:{Name:'W'}:false |
orderBy:['-Vorname', 'PassagierStatus'] | limitTo:10 track by
$index"> [...] </tr>
```

### Benutzerdefinierte Filter

```
var app = angular.module("Flug", []);
[...]
app.filter('globalize', function () {
    return function (input, format) {
        return Globalize.format(input, format);
    }
});
```

## Formulare

### Eigenschaft an Eingabefeld binden

```
<input ng-model="vm.flugNummerFilter" />
```

### Bool an Checkbox binden

```
<input type="checkbox" ng-model="vm.flug.verspaetet" />
```

### Optionen via Radio-Buttons anbieten

```
<input type="radio" ng-model="vm.flug.airline" name="airline"
value="LH" /> Lufthansa <br/>
<input type="radio" ng-model="vm.flug.airline" name="airline"
value="AUA" /> Austrian <br />
```

### Radio-Buttons aus Array erzeugen

```
<div ng-repeat="a in vm.airlines">
    <input type="radio" ng-model="vm.flug.airline"
        name="airline" value="{{a.id}}" /> {{ a.name }}
</div>
```

# Kurzreferenz („Cheat Sheet“) AngularJS

Autoren: FH-Prof. Manfred Steyer (M.Steyer@IT-Visions.at), Dr. Holger Schwichtenberg (H.Schwichtenberg@IT-Visions.de)

## Dropdown-Felder

```
<select ng-model="vm.flug.airline" ng-options="a.id as a.name  
for a in vm.airlines"></select>
```

## Dropdown-Felder mit Gruppierung

```
<select ng-model="vm.flug.airline" ng-options="a.id as a.name  
group by a.allianz for a in vm.airlines"></select>
```

## Validieren

### Auf Fehler in Formular prüfen

```
<form novalidate ng-controller="editFlugCtrl" name="form">
```

```
    <div>FlugNummer</div>
    <div><input ng-model="vm.flugNummer"
        name="flugNummer"/></div>

    <div>
        <b>Dirty (form):</b> {{ form.$dirty }}
    </div>
    <div>
        <b>Pristine (form):</b> {{ form.$pristine }}
    </div>
</form>
```

### Ausgewählte Validatoren

```
<form ng-controller="editFlugCtrl" name="form" novalidate>
    <div>FlugNummer</div>
    <div><input ng-model="vm.flugNummer" name="flugNummer"
        required ng-minlength="6" ng-maxlength="10"
        ng-pattern="/\w+\d+/" /></div>

    <div ng-show="form.flugNummer.$dirty &&
        form.flugNummer.$invalid">
        Flugnummer ist nicht gültig!
    </div>
    <div ng-show="form.flugNummer.$dirty &&
        form.flugNummer.$error.required">
        Pflichtfeld!
    </div>
</form>
```

### Zahlen validieren

```
<div>Preis</div>
<div><input ng-model="vm.preis" name="preis" type="number"
min="100" max="2000" /></div>
<div ng-show="form.preis.$dirty && form.preis.$invalid">
    Preis muss zwischen 0 und 2000 liegen!
</div>
```

### CSS-Klassen für Validierung

```
input.ng-invalid.ng-dirty {
    background-color: red;
}
input.ng-valid.ng-dirty {
    background-color: green;
}
```

### Benutzerdefinierte Validatoren

```
app.directive('integer', function () {
    return {
        require: 'ngModel',
        link: function (scope, elm, attrs, ctrl) {
            var format = attrs.integer;
            ctrl.$parsers.unshift(function (viewValue) {
                var integer = Globalize.parseInt(viewValue);
                if (!isNaN(integer)) {
                    ctrl.$setValidity('integer', true);
                    return integer;
                }
                else {
                    ctrl.$setValidity('integer', false);
                    return undefined;
                }
            });
            ctrl.$formatters.unshift(function (value) {
                debugger;
                var formatted = Globalize.format(value, format);
                return formatted;
            });
        }
    };
});
```

## Routing

```
<script src="~/Scripts/angular.js"></script>
<script src="~/Scripts/angular-route.js"></script>
```

```
<script>
    var app = angular.module("routing", ['ngRoute']);
    app.config(['$routeProvider', function ($routeProvider) {
        $routeProvider.
            when('/passagiere', {
                templateUrl: '/partials/routing.list.html',
                controller: 'ListCtrl'
            })
            .when('/passagiere/info', {
                templateUrl: '/partials/routing.info.html',
                controller: 'InfoCtrl'
            })
            .when('/passagiere/:id', {
                templateUrl: '/partials/routing.details.html',
                controller: 'DetailCtrl'
            })
            .otherwise({
                redirectTo: '/passagiere'
            });
    }]);

    app.controller("DetailCtrl", ['$scope', "$routeParams",
        function ($scope, $routeParams) {
            $scope.id = $routeParams.id;
        }
    ]);
</script>
```

### Platzhalter für View

```
<div ng-view></div>
```

### Link inkl. Routing-Parameter

```
<td><a href="#/routing/{{p.id}}">Details</a></td>
```

## Links

<https://angularjs.org/>

<http://www.IT-Visions.de/Schulung/AngularJS/7437>