

INHALTSVERZEICHNIS

Orderm8 – Website Technisches Handbuch	2
MVC.....	2
View.....	2
Controller	2
Model	2
Navigation	3
Webservice Layer	4

ORDERM8 – WEBSITE TECHNISCHES HANDBUCH

Die Website ist mit dem JavaScript Framework AngularJS implementiert worden. AngularJS automatisiert viele Vorgänge und ist im Gegensatz zu üblichen Websites übersichtlicher, da die Anzahl der Codezeilen reduziert wird.

MVC

VIEW

Die View ist wie gewohnt für die Darstellung zuständig. In diesem Falle war die View die HTML-Templates.

```
1 <div class="loginBackground">
2   <div class="login-area">
3     <h1>Login</h1>
4     <form name="formLogin">
5       <label>Username: </label>
6       <input name="username" type="text" id = "credentials" ng-model="credentials.username" required>
7       <label>Password:</label>
8       <input name="password" type="password" id = "password" ng-model="credentials.password" required>
9       <input id="btnLogin" class="btn btn-danger" type="submit" ng-disabled="!formLogin.$valid" ng-click="login()"
10        value="Log in">
11     </form>
12   </div>
13 </div>
```

CONTROLLER

Ist ein HTML-DOM-Element mit einer ng-controller Direktive ausgestattet, wird der Angular - Applikation ein Controller hinzugefügt, welcher innerhalb des Elements gültig ist.

Die Hauptaufgabe eines Controllers liegt im Data - Binding der Models an die Views.

Jeder Controller hat sein eigenes Scope-Objekt, auf welches außerhalb des Controller-Bereiches nicht zugegriffen werden kann. Mittels eines Angular-Controllers kann somit sichergestellt werden, dass Daten nur für jene Elemente zur Verfügung gestellt werden, die diese benötigen.

Des Weiteren ist es möglich, Controller zu schachteln, d.h. ein Controller ist innerhalb eines anderen Controllers. Das hat den Vorteil, dass das Scope-Objekt des äußeren Controllers nach innen hin sichtbar und veränderbar ist.

```
1 angular.module('login', [])
2
3 .controller('loginCtrl', ['$scope', 'userService', '$location', function($scope, userService, $location){
4   $scope.login = function(){
5     $scope.credentials = {};
6     $scope.credentials.username = "username";
7     $scope.credentials.password = "password";
8     var promiseGet = userService.promiseLogin($scope.credentials);
9     promiseGet.then(function(response){
10       $location.path('/overviewView')
11     }, function(response){
12       alert('login failed');
13     })
14   }
15 })
```

MODEL

Das Model beinhaltet Daten für den Controller damit dieser sie anzeigen kann, sowie Daten, die von Eingaben in der View stammen. Es wird dem Controller bei dessen Erstellung in Form von dem `$scope`-Objekt mitgegeben.

Im Screenshot vom Controller werden bei dessen Erstellung Defaultwerte für *username* & *password* gesetzt. Ist ein *input*-Field mit einer *ng-model*-Direktiven versehen, ändert sich automatisch das Model bei der Eingabe

durch den Benutzer. Werden im View-Screenshot die Credentials eingegeben, ändert sich das Model sofort. Es besteht also ein Two-Way-Data-Binding.

NAVIGATION

Mit AngularJS kann man einfach zwischen unterschiedlichen Templates navigieren. Dafür kann man eine Konfiguration mit einem `$routeProvider` festlegen.

```
app.config(function ($routeProvider) {

    $routeProvider
        .when('/waiterView', {
            templateUrl: 'templates/waiter.html',
            controller: 'waiterCtrl'
        })
        .when('/organiserView', {
            templateUrl: 'templates/organiser.html',
            controller: 'organiserCtrl'
        })
        .when('/overviewView', {
            templateUrl: 'templates/overview.html',
            controller: 'overviewCtrl'
        })
        .when('/tableView', {
            templateUrl: 'templates/table.html',
            controller: 'tableCtrl'
        })
        .when('/typeView', {
            templateUrl: 'templates/type.html',
            controller: 'typeCtrl'
        })
        .when('/productView', {
            templateUrl: 'templates/product.html',
            controller: 'productCtrl'
        })
        .when('/spatialView', {
            templateUrl: 'templates/spatial.html',
            controller: 'spatialCtrl'
        })
        .when('/loginView', {
            templateUrl: 'templates/login.html',
            controller: 'loginCtrl'
        })
        .otherwise('/loginView');
});
```

WEBSERVICE LAYER

Die Applikation benötigt unterschiedliche Webservices. Die dafür benötigten Aufrufe wurden von AngularJS-Factories übernommen. Während in der klassischen objektorientierten Programmierung eine Factory zum Erzeugen von Objekten verwendet wird, beinhaltet in AngularJS eine Factory wichtige Funktionen für Controller zur Datenbeschaffung.

```
.factory('statisticService', ['$http', '$q', 'URL', '$window', function($http, $q, URL, $window){
  var service = {
    promiseGetStatistic : promiseGetStatistic
  }
  return service;

  function promiseGetStatistic(){
    return $q(function(resolve, reject){
      $http({
        method: 'GET',
        url: URL + '/statistic',
        headers: {
          'x-access-token': $window.localStorage.getItem('token')
        }
      }).then(function successCallback(response) {
        resolve(response.data);
      }, function errorCallback(response) {
        reject(response);
      });
    });
  }
})
})
```