Web UI - Dev

Zadania do wykonania:

1. Pobierz i zainstaluj Nodejs: <https://nodejs.org/en/>
2. Dodaj lub zmodyfikuj zmienne środowiskowe:
   1. Przejdź do ‘Panel sterowania\System i zabezpieczenia\System’
   2. Wybierz ‘Zaawansowane ustawienia systemu’
   3. Wybierz ‘Zmienne środowiskowe’
   4. Dodaj lub zmodyfikuj ‘Path’, wartości: C:\Program Files\nodejs\
3. Otwórz Git Bash i stwórz na dysku C: strukturę katalogów:
   1. TechnicznyTester -> web -> app -> css,
   2. TechnicznyTester -> web -> app -> js,
   3. TechnicznyTester -> web -> app -> pages
4. Do katalogu TechnicznyTester -> web -> app dodaj trzy pliki:
   1. package.json - wypełnij zawartością podaną w części “Podpowiedzi”
   2. bower.json - wypełnij zawartością podaną w części “Podpowiedzi”
   3. Index.html
5. Otwórz konsolę cmd i wykonaj następujące czynności:
   1. Przejdź do katalogu ‘app’
   2. Wykonaj komendę “npm start” - to spowoduje instalację angularjs i wystartowanie serwera aplikacyjnego nodejs na określonym porcie i adresie. Wiele błędów i warningów może pojawić się w trakcie odpalania, nie przejmuj się to normalne. Bower zainstaluje wszystkie pakiety przez Ciebie podane.
6. W ktalogu TechnicznyTester -> web -> app -> js stwórz plik env.js gdzie będziesz przechowywać m.in. adres backendu do komunikacji

*//DEV Environment*

(**function** (window) {

window.\_\_env = window.\_\_env || {};

*// API url*

window.\_\_env.apiUrl = **'http://localhost:5001'**;

window.\_\_env.apiUrlService = **'http://localhost\\:5001'**;

window.\_\_env.enableDebug = **true**;

}(**this**));

1. Nasz frontend będzie potrzebował dobrej strony startowej w pliku index.html, na której będzie jeden jedyny przycisk [Start], po kliknięciu którego zostaniemy przeniesieni na stronę z listą kandydatów
2. Dodajmy także plik list.html do katalogu pages, w którym będziemy prezentowali listę kandydatów. W pliku tym także zdefiniujemy sobie okienko modalne do dodawania nowych kandydatów
3. Potrzebny nam będzie plik app.js w katalogu js. Plik ten będzie odpowiedzialny za przekierowania. Jeśli użytkownik kliknie np. Na przycisk [Start] to przekieruje nas to na stronę list.html.
4. Dodajmy także plik start.html do katalogu pages z przyciskiem [Start] żeby obsłużyć przypadek gdy cofamy się z listy na główną stronę
5. Nasz frontend będzie potrzebował sposobu komunikacji z backendem. Obsługiwane będzie to z poziomu pliku services.js w katalogu js. Zawierać on będzie informacje o endpointach i metodach dostarczonych przez backend.
6. Całość czyli HTML i serwices.js spina plik controller.js w katalogu js. Tutaj będziemy obsługiwali/wyciągali rzeczy z jsona i metody GET, a także tworzyli jsona dla metody POST
7. Na koniec dodajmy do katalogu css jakieś fajne css-y zawarte w pliku app.css

Podpowiedzi:

1. N/A
2. N/A
3. N/A
4. N/A
5. package.json

{

**"version"**: **"0.0.0"**,

**"private"**: **true**,

**"name"**: **"angular"**,

**"description"**: **"TechnicznyTester"**,

**"license"**: **"MIT"**,

**"devDependencies"**: {

**"bower"**: **"^1.3.1"**,

**"http-server"**: **"^0.9.0"**,

**"karma"**: **"^0.12.16"**,

**"karma-chrome-launcher"**: **"^0.1.4"**,

**"karma-firefox-launcher"**: **"^0.1.3"**,

**"karma-jasmine"**: **"~0.1.0"**,

**"protractor"**: **"^2.1.0"**,

**"shelljs"**: **"^0.2.6"**,

**"tmp"**: **"0.0.23"**

},

**"scripts"**: {

**"postinstall"**: **"bower install"**,

**"prestart"**: **"npm install"**,

**"start"**: **"http-server -a 0.0.0.0 -p 8000"**,

**"pretest"**: **"npm install"**,

**"test"**: **"node node\_modules/karma/bin/karma start test/karma.conf.js"**,

**"test-single-run"**: **"node node\_modules/karma/bin/karma start test/karma.conf.js --single-run"**,

**"preupdate-webdriver"**: **"npm install"**,

**"update-webdriver"**: **"webdriver-manager update"**,

**"preprotractor"**: **"npm run update-webdriver"**,

**"protractor"**: **"protractor test/protractor-conf.js"**,

**"update-index-async"**: **"node -e \"require('shelljs/global'); sed('-i', /\\/\\/@@NG\_LOADER\_START@@[\\s\\S]\*\\/\\/@@NG\_LOADER\_END@@/, '//@@NG\_LOADER\_START@@\\n' + cat('bower\_components/angular-loader/angular-loader.min.js') + '\\n//@@NG\_LOADER\_END@@', 'app/index-async.html');\""**

}

}

bower.json

{

**"name"**: **"angular-seed"**,

**"description"**: **"TechnicznyTester frontend"**,

**"version"**: **"0.0.0"**,

**"homepage"**: **"https://github.com/angular/angular-seed"**,

**"license"**: **"MIT"**,

**"private"**: **true**,

**"dependencies"**: {

**"angular"**: **"1.4.x"**,

**"angular-mocks"**: **"1.4.x"**,

**"jquery"**: **"~2.1.3"**,

**"bootstrap"**: **"~3.3.2"**,

**"angular-route"**: **"1.4.x"**,

**"angular-resource"**: **"1.4.x"**,

**"angular-animate"**: **"1.4.x"**,

**"angular-smart-table"**: **""**,

**"angularjs-datetime-picker"**: **""**,

**"moment"**: **""**,

**"angular-modal-service"**: **"0.6.10"**,

**"angular-ui-bootstrap-bower"**: **"1.3.2"**,

**"ng-csv"**: **"0.3.6"**,

**"angular-bind-html-compile"**: **""**,

**"angular-file-upload"**: **""**,

**"kineticjs"**: **""**

},

**"resolutions"**: {

**"angular"**: **"~1.4.x"**

}

}

1. NA
2. index.html

<!doctype **html**>

<**html lang="en" ng-app="TechnicalTester"**>

<**head**>

<**meta charset="utf-8"**>

<**title**>TechnicalTester</**title**>

<**link rel="stylesheet" href="css/app.css"**>

<**link rel="stylesheet" href="bower\_components/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet"**>

<**script src="bower\_components/jquery/dist/jquery.js"**></**script**>

<**script src="bower\_components/bootstrap/dist/js/bootstrap.js"**></**script**>

<**script src="bower\_components/angular/angular.js"**></**script**>

<**script src="bower\_components/angular-animate/angular-animate.js"**></**script**>

<**script src="bower\_components/angular-route/angular-route.js"**></**script**>

<**script src="bower\_components/angular-resource/angular-resource.js"**></**script**>

<**script src="bower\_components/angular-modal-service/dst/angular-modal-service.js"**></**script**>

<**script src="js/env.js"**></**script**>

<**script src="js/app.js"**></**script**>

<**script src="js/controller.js"**></**script**>

<**script src="js/services.js"**></**script**>

<**script src="//ajax.googleapis.com/ajax/libs/angularjs/1.4.0/angular-messages.js"**></**script**>

<**script src="bower\_components/angular-ui-bootstrap-bower/ui-bootstrap-tpls.js"**></**script**>

<**script src="bower\_components/angular-smart-table/dist/smart-table.js"**></**script**>

</**head**>

<**body ng-controller="mainCtrl"**>

<**div id="main"**>

<**div ng-view**></**div**>

</**div**>

</**body**>

</**html**>

ng-view to kontrolka, która nam wstrzykuje zawartość html’a

1. List.html

<**body ng-app = "TechnicalTester" ng-controller="listCtrl"**>

<**a href="#start" class="btn btn-primary"**>Back</**a**>

<**div ng-controller="tableCtrl"**>

<**div ng-controller="ModalCtrl"**>

*<!--Add Candidate Modal-->*

<**script type="text/ng-template" id="newModalContent.html"**>

<**div class="modal-header"**>

<**button type="button" class="close" ng-click="cancel()" data-dismiss="modal" aria-label="Close"**><**span aria-hidden="true"**>**&times;**</**span**></**button**>

<**h3 class="modal-title"**>Add Candidate</**h3**>

</**div**>

<**table class="main"**>

<**tr**>

<**td style="width:60%;vertical-align: top"**>

<**form name="addCandidate" class="form-horizontal" role="form" style="width:100%;margin-left:30px" novalidate ng-submit="addCandidate.$valid && addRow()"**>

<**div class="modal-body" style="margin-top:-70px"**>

<**div class="form-group"**>

<**label class="col-md-4 control-label"**>Name</**label**>

<**div class="col-md-4"**>

<**input type="text" class="form-control" name="name" ng-model="name" ng-maxlength="10" placeholder="e.g. John" required**/>

<**div ng-messages="addCandidate.name.$error" style="color:maroon" role="alert" ng-if='submitted'**>

<**div ng-message="required"**>This field is required</**div**>

<**div ng-message="maxlength"**>This field is too long</**div**>

</**div**>

</**div**>

</**div**>

<**div class="form-group"**>

<**label class="col-md-4 control-label"**>Last Name</**label**>

<**div class="col-md-4"**>

<**input type="text" class="form-control" name="lastName" ng-model="lastName" ng-maxlength="10" placeholder="e.g. Rambo" required**/>

<**div ng-messages="addCandidate.lastName.$error" style="color:maroon" role="alert" ng-if='submitted'**>

<**div ng-message="required"**>This field is required</**div**>

<**div ng-message="maxlength"**>This field is too long</**div**>

</**div**>

</**div**>

</**div**>

<**div class="form-group"**>

<**label class="col-md-4 control-label"**>Level</**label**>

<**div class="col-md-4"**>

<**input type="text" class="form-control" name="level" ng-model="level" ng-maxlength="10" required**/>

<**div ng-messages="addCandidate.level.$error" style="color:maroon" role="alert" ng-if='submitted'**>

<**div ng-message="required"**>This field is required</**div**>

<**div ng-message="maxlength"**>This field is too long</**div**>

</**div**>

</**div**>

</**div**>

<**div class="form-group"**>

<**label class="col-md-4 control-label"**>Area</**label**>

<**div class="col-md-4"**>

<**input type="text" class="form-control" name="area" ng-model="area" ng-maxlength="10" required**/>

<**div ng-messages="addCandidate.area.$error" style="color:maroon" role="alert" ng-if='submitted'**>

<**div ng-message="required"**>This field is required</**div**>

<**div ng-message="maxlength"**>This field is too long</**div**>

</**div**>

</**div**>

</**div**>

<**div class="form-group"**>

<**label class="col-md-4 control-label"**>Salary</**label**>

<**div class="col-md-4"**>

<**input type="text" class="form-control" name="salary" ng-model="salary" ng-maxlength="10" required**/>

<**div ng-messages="addCandidate.salary.$error" style="color:maroon" role="alert" ng-if='submitted'**>

<**div ng-message="required"**>This field is required</**div**>

<**div ng-message="maxlength"**>This field is too long</**div**>

</**div**>

</**div**>

</**div**>

<**div class="form-group"**>

<**label class="col-md-4 control-label"**>Grade</**label**>

<**div class="col-md-4"**>

<**input type="text" class="form-control" name="grade" ng-model="grade" ng-maxlength="10" required**/>

<**div ng-messages="addCandidate.grade.$error" style="color:maroon" role="alert" ng-if='submitted'**>

<**div ng-message="required"**>This field is required</**div**>

<**div ng-message="maxlength"**>This field is too long</**div**>

</**div**>

</**div**>

</**div**>

<**div class="form-group"**>

<**label class="col-md-4 control-label"**>CV</**label**>

<**div class="col-md-4"**>

<**input type="file" class="btn btn-default" file-model = "myFile" name="uFileName" accept="text/csv"**/>

</**div**>

</**div**>

</**div**>

<**div class="modal-footer" style="padding-right:100px;border-top: 0px;"**>

<**input type="submit" value="Submit" class="btn btn-primary" ng-click="submitted=true" ng-keyup="submitted=false"**/>

<**button type="button" ng-click="cancel()" class="btn btn-default" data-dismiss="modal"**>Close</**button**>

</**div**>

</**form**>

</**td**>

</**tr**>

</**table**>

</**script**>

*<!--Candidates Table-->*

<**table st-table="displayedData" st-safe-src="data" class="tableList"**>

<**tr**>

<**td style="**width:10%;vertical-align: top;border-bottom: 0px;**"**><**input type="button" value="Add Candidate" class="btn btn-primary" ng-click="addCandidate()"**/></**td**>

</**tr**>

<**tr**>

<**th st-sort="name"**>Name</**th**>

<**th style="**width:15%**" st-sort="lastName"**>Last Name</**th**>

<**th**>Accepted</**th**>

</**tr**>

<**tr ng-repeat="data in data"** >

<**td style="**text-align: left;**"**>{{data.name}}</**td**>

<**td**>{{data.lastName}}</**td**>

<**td**>{{data.accepted}}</**td**>

</**tr**>

</**table**>

</**div**>

</**div**>

</**body**>

1. app.js

**'use strict'**;

*// create the module and name it TechnicalTester*

angular.module(**'TechnicalTester'**, [**'ngRoute'**, **'TechnicalTester.services'**, **'TechnicalTester.controller'**, **'ngMessages'**])

*// configure routes*

.config([**'$routeProvider'**, **'$locationProvider'**, **function**($routeProvider, $locationProvider) {

$routeProvider

.when(**'/'**, {

templateUrl : **'pages/start.html'**,

controller : **'listCtrl'**

})

.when(**'/list'**, {

templateUrl : **'pages/list.html'**,

controller : **'listCtrl'**

})

.when(**'/start'**, {

templateUrl : **'pages/start.html'**,

controller : **'startCtrl'**

})

}]);

1. Dodaj pliczek start.html
2. services.js

**'use strict'**;

**var** services = angular.module(**'TechnicalTester.services'**, [**'ngResource'**]);

*//Table list data*

services.factory(**'getData'**, [**'$resource'**,

**function** ($resource) {

**return** $resource(\_\_env.apiUrlService+**'/data'**, {}, {

query: {method:**'GET'**, isArray:**true**}

});

}]);

*//Add new candidate*

services.factory(**'postCandidate'**, [**'$resource'**,

**function** ($resource) {

**return** $resource(\_\_env.apiUrlService+**'/addCandidate'**, {}, {

save: {method:**'POST'**, isArray:**true**}

});

}]);

1. controller.js

**'use strict'**;

**var** TechnicalTester = angular.module(**'TechnicalTester.controller'**, [**'ngResource'**, **'ngAnimate'**, **'ui.bootstrap'**, **'smart-table'**,

**'angularModalService'**]);

*// Register environment in AngularJS as constant*

TechnicalTester.constant(**'\_\_env'**, \_\_env);

*//-----------------------------------------------------Start------------------------------------------------------------*

*//startCtrl*

TechnicalTester.controller(**'startCtrl'**, **function**($scope) {

});

*//-----------------------------------------------------List-------------------------------------------------------------*

*//listCtrl*

TechnicalTester.controller(**'listCtrl'**, **function**($scope) {

});

*//tableCtrl*

TechnicalTester.controller(**'tableCtrl'**, [**'$rootScope'**, **'$routeParams'**, **'getData'**,

**function**($scope, $routeParams, Data) {

*//Get data*

$scope.displayedData = [];

$scope.data = Data.query();

console.log($scope.data);

}]);

*//ModalCtrl*

TechnicalTester.controller(**'ModalCtrl'**, [**'$scope'**, **'$uibModal'**, **function** ($scope, $uibModal) {

$scope.animationsEnabled = **true**;

*//Add Candidate Modal*

$scope.addCandidate = **function** () {

**var** modalInstance = $uibModal.open({

animation: $scope.animationsEnabled,

templateUrl: **'newModalContent.html'**,

controller: **'ModalInstanceCtrl'**,

});

};

}]);

TechnicalTester.controller(**'ModalInstanceCtrl'**, [**'$rootScope'**, **'$scope'**, **'$uibModalInstance'**, **'postCandidate'**,

**function** ($rootScope, $scope, $uibModalInstance, NewCandidate) {

*//[Submit]*

$scope.addRow = **function**(){

$rootScope.displayedData = [];

$rootScope.data = [];

*//Call postCandidate service...*

NewCandidate.save({name:$scope.name, lastName:$scope.lastName, level:$scope.level,

area:$scope.area, salary:$scope.salary, grade:$scope.grade}, **function**(response){

$scope.message = response.message;

**var** globalObject = [];

**var** len = response.length;

**for**(**var** i = 0; i < len; i++) {

**var** newObject = {}

angular.forEach(response[Object.keys(response)[i]], **function**(value, key){

newObject[key] = value;

});

globalObject.push(newObject);

}

$rootScope.displayedData = [];

$rootScope.data = globalObject;

});

*//...and close modal*

$uibModalInstance.dismiss();

};

*//[Cancel]*

$scope.cancel = **function** () {

$uibModalInstance.dismiss(**'cancel'**);

};

}]);

*//-----------------------------------------------------------Home-------------------------------------------------------*

*//mainCtrl*

TechnicalTester.controller(**'mainCtrl'**, [**'$rootScope'**, **function** ($scope) {

}]);

1. app.css

*/\* app css stylesheet \*/*

.objects {

list-style: none;

}

body{

padding-top: 20px;

font:15px/1.3 **'Open Sans'**, sans-serif;

color: #5e5b64;

text-align:center;

}

a, a:visited {

outline:none;

color:#389dc1;

}

a:hover{

text-decoration:none;

}

.active {

background-color: #4CAF50;

}

p{

font-size:22px;

font-weight:bold;

color:white;

}

p b{

color:white;

display:inline-block;

padding:5px 10px;

background-color:#c4d7e0;

border-radius:2px;

text-transform:uppercase;

font-size:18px;

}

.main {

border-collapse: collapse;

width: 100%;

max-width: 1100px;

margin: 100px auto;

}

.main tr {

background-color:white

}

.main th {

width: 100%;

padding: 8px;

text-align: center;

border-bottom: 1px solid #ddd;

background-color:#f5f5f5

}

.main td {

width: 100%;

padding: 8px;

text-align: center;

border-bottom: 1px solid #ddd;

}

.tableList {

border-collapse: collapse;

width: 100%;

max-width: 1100px;

margin: 20px 20px 20px 20px;

display:inline-block;

}

.tableList tr {

background-color:white

}

.tableList th {

width: 10%;

padding: 8px;

text-align: center;

border-bottom: 1px solid #ddd;

background-color:#f5f5f5

}

.tableList td {

width: 10%;

padding: 5px;

text-align: center;

border-bottom: 1px solid #ddd;

}

Rozwiązanie:

Pełne rozwiązanie znajduje się w pliku App.7z