ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент, к.т.н. |  |  |  | П.А. Степанов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

­­

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| РАЗРАБОТКА ПРОСТОГО AJAX ПРИЛОЖЕНИЯ SPRING |
| по дисциплине: ТЕХНОЛОГИИ РАЗРАБОТКИ СЕРВЕРНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4032 |  |  |  | М.В. Мухортов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Задание**

1. Создайте одну или несколько форм и контроллеров для страниц приложения. Подключите библиотеку angularJS (желающие могут воспользоваться JQuery или иным аналогом).
2. Реализуйте задачи из лабораторной работы 1 средствами AJAX и REST-сервисов, разработанных в предыдущей лабораторной работе.

Вариант7: Ведение списка литературы согласно последнему актуальному ГОСТу

**Описание разрабатываемого продукта**

В ходе выполнения работы необходимо разработать серверное приложение с реализацией REST контроллера средствами Spring-boot, а также использованием технологии AJAX для выполнения операций просмотра, добавления, удаления и вставки (изменения) литературы в список.

**Текст основных фрагментов кода**

**literature.html**

<!DOCTYPE html>  
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org"  
 xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">   
 <head>  
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"></meta>  
 <title>Список литературы!</title>  
 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.3/angular.min.js"></script>  
 <script src="js/literature.js"></script>  
  
 </head>  
 <body ng-app="literature" ng-controller="LiteratureController">  
  
 <h1>Список литературы</h1>  
  
 <div ng-controller="getLiterature" ng-show="literatureList.length > 0">  
 <table id="literatureTable">  
 <thead>  
 <tr>  
 <th>ИД</th>  
 <th>Значение</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr ng-repeat="literature in literatureList" id="{{literature.id}}">  
 <td>{{literature.id}}</td>  
 <td>{{literature.source}}</td>  
 <td><input type="button" value="Удалить" ng-click="delete(literature.id)"/></td>  
  
 </tr>  
 </tbody>  
 </table>  
 </div>  
 <div>  
 <form role="form" ng-submit="addSource()">  
 <div>  
 <label>Значение</label>  
 <div>  
 <input type="text" name="source" ng-model="source" />  
 </div>  
 </div>  
 <div>   
 <div>  
 <input type="submit" value="Добавить"/>  
 </div>  
 </div>  
 </form>  
 </div>  
 <div>  
 <form role="form" ng-submit="putSource(id)">  
 <div>  
 <label>Ид</label>  
 <div>  
 <input type="number" name="id" ng-model="id" />  
 </div>  
 </div>  
 <div>  
 <label>Значение</label>  
 <div>  
 <input type="sourcePatch" name="id" ng-model="sourcePatch" />  
 </div>  
 </div>  
 <div>  
 <input type="submit" value="Изменить" />  
 </div>  
 </form>  
 </div>  
 </body>  
</html>

**literature.js**

var ***app*** = angular.module('literature', []);  
  
***app***.controller("LiteratureController", function ($scope, $http) {  
  
 $scope.getLiterature = function () {  
 $http.get('/public/rest/literature').success(function (data, status, headers, config) {  
 $scope.literatureList = data;  
 for (var i = 0; i < $scope.literatureList.length; i++) {  
 $scope.literatureList[i].edit = 0;  
 }  
 }).error(function (data, status, headers, config) {  
 if (data.message === 'Time is out') {  
 $scope.finishByTimeout();  
 }  
 });  
 };  
  
 $scope.delete = function (id) {  
 $http.delete('/public/rest/literature/' + id).success(function (data, status, headers, config) {  
 for (var i = 0; i < $scope.literatureList.length; i++) {  
 var j = $scope.literatureList[i];  
 if (j.id === id) {  
 $scope.literatureList.splice(i, 1);  
 break;  
 }  
 }  
 }).error(function (data, status, headers, config) {  
 ***console***.error(status, data, headers);  
 });  
 };  
  
 $scope.addSource = function () {  
 $http.post('/public/rest/literature/' + $scope.source).success(function (data, status, headers, config) {  
 $scope.literatureList.splice($scope.literatureList.length, 0, data);  
 }  
 ).error(function (data, status, headers, config) {  
 ***console***.error(status, data, headers);  
 });  
  
 };  
  
 $scope.putSource = function (id) {  
 $http.put('/public/rest/literature/' + $scope.id + "/" + $scope.sourcePatch).success(function (data, status, headers, config) {  
 for (var i = 0; i < $scope.literatureList.length; i++) {  
 var m = $scope.literatureList[i];  
 if (m.id === id) {  
 $scope.literatureList.splice(i, 1, data);  
 break;  
 }  
 if (i === $scope.literatureList.length - 1) {  
 $scope.literatureList.splice(i + 1, 0, data);  
 break;  
 }  
 }  
 }  
 ).error(function (data, status, headers, config) {  
 ***console***.error(status, data, headers);  
 });  
 }  
});