ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| кандидат технических наук |  |  |  | П. А. Степанов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3 |
| Разработка простого AJAX приложения Spring |
| по курсу: Технология разработки серверных информационных систем |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134к |  |  |  | Столяров Н.С. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

**Текст и вариант задания:**

19 вариант

База данных с рецептами блюд

**Описание разрабатываемого продукта:**

В данной программе создан сервис по работе с рецептами блюд (вывод, добавление, удаление и поиск)

**Текст основных фрагментов кода :**

**ReceptRestService.java**

package info.stepanoff.trsis.samples.rest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RestController;

import info.stepanoff.trsis.samples.service.ReceptService;

import lombok.extern.slf4j.Slf4j;

@Slf4j

@RestController

@RequestMapping("/public/rest/Recept")

public class ReceptRestService {

@Autowired

private ReceptService receptService;

@RequestMapping(value = "", method = RequestMethod.GET)

public ResponseEntity<Object> browse() {

return ResponseEntity.ok(receptService.listAll());

}

@RequestMapping(value = "/{id}", method = RequestMethod.DELETE)

public void delete(@PathVariable("id") Integer id) {

receptService.delete(id);

}

@RequestMapping(value = "/{dificulty}/{name}/{\_body}", method = RequestMethod.POST)

public ResponseEntity<Object> add(

@PathVariable("dificulty") int \_dificulty,

@PathVariable("name") String name,

@PathVariable("\_body") String \_body) {

return ResponseEntity.ok(receptService.add(\_dificulty, name, \_body));

}

//int \_quantity, String \_name, String \_expiration\_date

@RequestMapping(value = "/{name}", method = RequestMethod.GET)

public ResponseEntity<Object> findBy\_name(@PathVariable("name") String name) {

return ResponseEntity.ok(receptService.findBy\_name(name));

}

}

**Recept.java**

package info.stepanoff.trsis.samples.db.model;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

import javax.persistence.\*;

import java.io.Serializable;

import java.util.List;

import java.util.Map;

@Entity

@Table(name = "RECEPT")

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Recept implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@Column(name = "RECEPT\_ID")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer id;

@OneToMany(targetEntity = Batch.class, mappedBy = "Recept", fetch = FetchType.LAZY)

private List<Batch> batchesList;

@Column(name = "RECEPT\_DIFICULTY", unique = true)

private int dificulty;

@Column(name = "RECEPT\_NAME")

private String name;

@Column(name = "RECEPT\_BODY")

private String body;

public Recept(int \_dificulty, String name, String \_body){

dificulty =\_dificulty;

this.name = name;

body = \_body;

}

}

**Recept.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 charset="utf-8"></meta>

<meta http-equiv="X-UA-Compatible" content="IE=edge"></meta>

<meta name="viewport" content="width=device-width, initial-scale=1"></meta>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"></meta>

<title>Schools in our university</title>

<link href="../static/css/bootstrap.min.css" th:href="@{/css/bootstrap.min.css}" rel="stylesheet"></link>

<script src="js/jquery-1.9.1.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/angular-1.7.5.min.js"></script>

<script src="js/Recept.js"></script>

<style>

html {

filter: invert(100%);

}

body {

padding-left: 100px;

padding-right: 100px;

}

h1 {

width: 100%;

text-align: center;

}

</style>

</head>

<body ng-app="Recept" ng-controller="ReceptController">

<h1>Список Рецептов</h1>

<div ng-controller="getRecept" ng-show="ReceptList.length > 0">

<table id="schoolsTable" class="table table-hover">

<thead>

<tr>

<th><h2>Сложность</h2></th>

<th><h2>Название</h2></th>

<th>Рецепт</th>

<th></th>

</tr>

</thead>

<tbody>

<tr ng-repeat="Recept in ReceptList" id="{{Recept.id}}">

<td>{{Recept.dificulty}}</td>

<td>{{Recept.name}}</td>

<td>{{Recept.body}}</td>

<!-- <td><input type="button" value="Редактировать"/></td> -->

<td><input type="button" value="Удалить" ng-click="deleteRecept(Recept.id)"/></td>

</tr>

</tbody>

</table>

</div>

<div class="alert alert-danger" role="alert" ng-show="errormessage">{{errormessage}}</div>

<div>

<form role="form" ng-submit="addRecept()">

<div class="form-group">

<label for="inputnumber">Номер</label>

<input type="number" class="form-control" id="inputnumber" name="inputnumber" placeholder="Номер" ng-model="inputnumber"> </input>

</div>

<div class="form-group">

<label for="inputname">Название</label>

<input type="text" class="form-control" id="inputname" name="inputname" ng-model="inputname" placeholder="Название"></input>

</div>

<div class="form-group">

<label for="inputbody">Рецепт</label>

<input type="text" class="form-control" id="inputbody" name="inputbody" ng-model="inputbody" placeholder="Рецепт"></input>

</div>

<button type="submit" class="btn btn-default">Добавить</button>

</form>

</div>

</body>

</html>

**Recept.js**

var app = angular.module('Recept', []);

app.controller("ReceptController", function ($scope, $http) {

$scope.successGetFridgsCallback = function (response) {

$scope.ReceptList = response.data;

$scope.errormessage="";

};

$scope.errorGetFridgsCallback = function (error) {

console.log(error);

$scope.errormessage="Unable to get list of Recept";

};

$scope.getRecept = function () {

$http.get('/public/rest/Recept').then($scope.successGetFridgsCallback, $scope.errorGetFridgsCallback);

};

$scope.successDeleteReceptCallback = function (response) {

for (var i = 0; i < $scope.ReceptList.length; i++) {

var j = $scope.ReceptList[i];

if (j.id === $scope.deletedId) {

$scope.ReceptList.splice(i, 1);

break;

}

}

$scope.errormessage="";

};

$scope.errorDeleteReceptCallback = function (error) {

console.log(error);

$scope.errormessage="Unable to delete Recept; check if there are any related records exist. Such records should be removed first";

};

$scope.deleteRecept = function (id) {

$scope.deletedId = id;

$http.delete('/public/rest/Recept/' + id).then($scope.successDeleteReceptCallback, $scope.errorDeleteReceptCallback);

};

$scope.successGetReceptCallback = function (response) {

console.log(response.data);

$scope.ReceptList.splice(0, 0, response.data);

// $scope.ReceptList.push(response.data);

$scope.errormessage="";

};

$scope.errorGetReceptCallback = function (error) {

console.log(error);

$scope.errormessage="Unable to get information on Recept number " + $scope.inputnumber;

};

$scope.successAddReceptCallback = function (response) {

console.log(response);

// $http.get('/public/rest/Recept/' + $scope.inputname).then($scope.successGetReceptCallback, $scope.errorGetReceptCallback);

// $scope.ReceptList.push({

// "dificulty": $scope.inputnumber,

// "name": $scope.inputname,

// "body": $scope.inputbody,

// "id": $scope.ReceptList.length + 1

// });

$scope.ReceptList.splice(0, 0, response.data);

$scope.errormessage="";

};

$scope.errorAddReceptCallback = function (error) {

console.log(error);

$scope.errormessage="Impossible to add new Recept; check if it's number is unique";

};

$scope.addRecept = function () {

$http.post('/public/rest/Recept/' + $scope.inputnumber + "/" + $scope.inputname + "/" + $scope.inputbody).then($scope.successAddReceptCallback, $scope.errorAddReceptCallback);

};

});