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 №2 |
| Разработка ресурса REST/JSON сервиса |
| 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 :**

package com.example.lb2;

import com.example.lb2.dto.Fridge;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.nio.file.Files;

import java.nio.file.Paths;

import java.nio.file.StandardOpenOption;

import java.io.\*;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

@SpringBootApplication

public class Lb2Application {

public static void main(String[] args) {

SpringApplication.run(Lb2Application.class, args);

}

public static void add(String name, String recept) {

if (name != null && !name.isEmpty() && recept != null && !recept.isEmpty()) {

try {

name = name + "\n";

recept = recept + "\n";

Files.write(Paths.get("base.txt"), name.getBytes(),StandardOpenOption.APPEND);

Files.write(Paths.get("base.txt"), recept.getBytes(), StandardOpenOption.APPEND);

}

catch (IOException e) {

System.out.println(e);

}

}

}

public static Map<String, String> get() {

int id = 0;

Map<String, String> result = new HashMap<String, String>();

try {

File file = new File("base.txt");

FileReader fr = new FileReader(file);

BufferedReader reader = new BufferedReader(fr);

String line = reader.readLine();

String recept;

while (line != null) {

recept = reader.readLine();

result.put(line, recept);

line = reader.readLine();

id++;

}

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

return result;

}

public static Boolean remove(String find\_name) throws Exception {

System.out.println(find\_name);

File sourceFile = new File("base.txt");

File outputFile = new File("base\_buff.txt");

boolean result = false;

BufferedReader reader = new BufferedReader(new FileReader(sourceFile));

BufferedWriter writer = new BufferedWriter(new FileWriter(outputFile));

String line = "";

boolean find = false;

while ((line = reader.readLine()) != null) {

if (find) {

find = false;

} else {

if (!line.contains(find\_name)) {

writer.write(line);

writer.newLine();

} else {

find = true;

result = true;

}

}

}

reader.close();

writer.close();

sourceFile.delete();

outputFile.renameTo(sourceFile);

return result;

}

}

package com.example.lb2.controller;

import com.example.lb2.Lb2Application;

import com.example.lb2.dto.Fridge;

import org.apache.coyote.Response;

import org.springframework.http.\*;

import org.springframework.web.bind.annotation.\*;

import org.springframework.web.client.HttpStatusCodeException;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import static java.lang.System.\*;

import static org.springframework.http.MediaType.APPLICATION\_JSON\_VALUE;

@RestController

public class restcontroller {

@GetMapping(path ="/list",produces = MediaType.APPLICATION\_JSON\_VALUE)

public ResponseEntity<Map<String,String>> getMethod(){

return ResponseEntity.ok(Lb2Application.get());

}

@DeleteMapping(path ="/list{name}",produces = MediaType.APPLICATION\_JSON\_VALUE)

public ResponseEntity<Map<String,Boolean>> deleteMethod(@PathVariable("name") String name) throws Exception {

boolean result = Lb2Application.remove(name);

System.out.println(result);

if(result){

Map<String,Boolean> result\_ok = new HashMap<String,Boolean>() {{

put("status", true);

}};

return ResponseEntity.ok(result\_ok);

}else{

Map<String,Boolean> result\_fail = new HashMap<String,Boolean>() {{

put("status", false);

}};

return ResponseEntity.ok(result\_fail);

}

}

@PostMapping(path ="/list",produces = MediaType.APPLICATION\_JSON\_VALUE)

public ResponseEntity<Map<String,Boolean>> PostMethod(@RequestBody Map<String, String> map\_data){

Map<String,Boolean> result = new HashMap<String,Boolean>() {{

put("status", true);

}};

Lb2Application.add(

map\_data.get("name"),

map\_data.get("recept")

);

return ResponseEntity.ok(result);

}

}