ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| РАЗРАБОТКА РЕСУРСА REST/JSON СЕРВИСА |
| по курсу: ТЕХНОЛОГИИ РАЗРАБОТКИ СЕРВЕРНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | 4136 |  |  |  | Н.С. Бобрович |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

**Задание (Вариант 3)**

Расписание занятий в институте

**Описание разрабатываемого продукта**

Программа представляет собой веб-приложение с возможностью просмотра, добавления и удаления занятий.

Текст программы

**openapi**

{

"openapi": "3.0.1",

"info": {

"title": "Расписание занятий студентов",

"version": "1.0.0"

},

"servers": [

{

"url": "http://localhost:8887/Lab1",

"description": "Dev server"

}

],

"paths": {

"/Lesson": {

"get": {

"summary": "Метод получения расписания",

"tags": [

"Lessons"

],

"operationId": "getAllLessons",

"responses": {

"200": {

"description": "Успешный ответ со списком занятий",

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/Lessons"

}

}

}

},

"404": {

"description": "Страница не найдена",

"content": {}

}

}

},

"post": {

"tags": [

"Lessons"

],

"summary": "Добавить новое занятие",

"operationId": "addLesson",

"requestBody": {

"required": true,

"description": "Добавление нового занятие",

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/Lesson"

}

}

}

},

"responses": {

"201": {

"description": "Успешная операция",

"content": {

"application/json": {

"schema": {

"type": "array",

"items": {

"$ref": "#/components/schemas/Lesson"

}

}

}

}

},

"400": {

"description": "Неверный формат ввода",

"content": {}

}

}

}

},

"/Lesson/{num\_group}": {

"delete": {

"summary": "Метод удаления занятия",

"tags": [

"Lessons"

],

"operationId": "deleteLesson",

"parameters": [

{

"name": "num\_group",

"in": "path",

"required": true,

"description": "Номер группы",

"schema": {

"type": "string"

},

"example": 4132

}

],

"responses": {

"200": {

"description": "Успешное удаления",

"content": {

"application/json": {}

}

}

}

}

}

},

"components": {

"schemas": {

"Lesson": {

"type": "object",

"required": [

"num\_group",

"audience",

"teacher",

"name",

"number"

],

"properties": {

"id": {

"type": "integer",

"example": 0

},

"num\_group": {

"type": "string",

"example": 4132

},

"audience": {

"type": "string",

"example": "23-09"

},

"teacher": {

"type": "string",

"example": "Ivanov"

},

"name": {

"type": "string",

"example": "Physics"

},

"number": {

"type": "integer",

"example": 5

}

}

},

"Lessons": {

"type": "array",

"items": {

"$ref": "#/components/schemas/Lesson"

}

}

}

}

}

**DBHelper**

package io.swagger;  
  
import io.swagger.model.Lesson;  
import io.swagger.model.Lessons;  
  
public class DBHelper {  
 static public Lessons *db* = new Lessons() {{  
 new Lesson().numGroup("4131").audience("23-08").teacher("Ivanov").name("History").number(3);  
 new Lesson().numGroup("4132").audience("23-10").teacher("Petrov").name("Physics").number(2);  
 new Lesson().numGroup("4134").audience("22-18").teacher("Smirnov").name("Math").number(5);  
 }};  
}

**LessonApiController**

package io.swagger.api;  
  
import io.swagger.DBHelper;  
import io.swagger.model.Lesson;  
import io.swagger.model.Lessons;  
import com.fasterxml.jackson.databind.ObjectMapper;  
import io.swagger.v3.oas.annotations.Parameter;  
import io.swagger.v3.oas.annotations.enums.ParameterIn;  
import io.swagger.v3.oas.annotations.media.Schema;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestBody;  
  
import javax.validation.Valid;  
import javax.servlet.http.HttpServletRequest;  
import java.io.IOException;  
import java.util.List;  
  
@javax.annotation.Generated(value = "io.swagger.codegen.v3.generators.java.SpringCodegen", date = "2023-09-13T19:36:10.460669586Z[GMT]")  
@RestController  
public class LessonApiController implements LessonApi {  
  
 private static final Logger *log* = LoggerFactory.*getLogger*(LessonApiController.class);  
  
 private final ObjectMapper objectMapper;  
  
 private final HttpServletRequest request;  
  
 @org.springframework.beans.factory.annotation.Autowired  
 public LessonApiController(ObjectMapper objectMapper, HttpServletRequest request) {  
 this.objectMapper = objectMapper;  
 this.request = request;  
 }  
  
 public ResponseEntity<List<Lesson>> addLesson(@Parameter(in = ParameterIn.*DEFAULT*, description = "Добавление нового занятие", required=true, schema=@Schema()) @Valid @RequestBody Lesson body) {  
 String accept = request.getHeader("Accept");  
 if (accept != null && accept.contains("application/json")) {  
 DBHelper.*db*.add(body);  
 return new ResponseEntity<List<Lesson>>(DBHelper.*db*, HttpStatus.*OK*);  
 }  
  
 return new ResponseEntity<List<Lesson>>(HttpStatus.*NOT\_FOUND*);  
 }  
  
 public ResponseEntity<Void> deleteLesson(@Parameter(in = ParameterIn.*PATH*, description = "Номер группы", required=true, schema=@Schema()) @PathVariable("num\_group") String numGroup) {  
 String accept = request.getHeader("Accept");  
 for(int i=0;i<DBHelper.*db*.size();++i){  
 if(DBHelper.*db*.get(i).getNumGroup().contentEquals(numGroup)){  
 DBHelper.*db*.remove(i);  
 break;  
 }  
 }  
 return new ResponseEntity<Void>(HttpStatus.*OK*);  
 }  
  
 public ResponseEntity<List<Lesson>> getAllLessons() {  
 String accept = request.getHeader("Accept");  
 if (accept != null && accept.contains("application/json")) {  
 return new ResponseEntity<List<Lesson>>((Lessons)DBHelper.*db*, HttpStatus.*OK*);  
 }  
  
 return new ResponseEntity<List<Lesson>>(HttpStatus.*NOT\_FOUND*);  
 }  
  
}