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 дисциплине: Технология разработки серверных информационных систем |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134к |  |  |  | Костяков Н.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

## Задача 1

Определите перечень Rest-сервисов, выполняющих те же действия, что и в лабораторной работе 1. Внимательно отнеситесь к вопросу какой HTTP метод использует тот или иной сервис и какие коды HTTP он может возвращать. 2 Опишите Ваш API с помощью OpenAPI v3. 3 Средствами swagger сгенерируйте сервер spring 4 Замените pom файл в сгенерированном сервере на файл из моего примера 5 Поменяйте названия пакетов на уникальные 6 Реализуйте приложение

## Листинг

package com.example.demo;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

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

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

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

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

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

*@RestController*

*@RequestMapping*("/api")

public class MyController {

public List<String> bus = new ArrayList<String>();

public List<String> plane = new ArrayList<String>();

public List<String> ships = new ArrayList<String>();

MyController(){

bus.add("123 - каждые 13 минут");

bus.add("125 - каждые 12 минут");

bus.add("127 - каждые 156 минут");

bus.add("129 - каждые 138 минут");

plane.add("3018 - каждые 2 часа");

plane.add("3016 - каждые 2 часа");

plane.add("30187 - каждые 2 часа");

plane.add("3011 - каждые 2 часа");

ships.add("wer - каждые 2 часа");

ships.add("rty - каждые 2 часа");

ships.add("dfg - каждые 2 часа");

ships.add("cvb - каждые 2 часа");

}

*@GetMapping*("/bus")

public ResponseEntity<Object> getBus() {

return new ResponseEntity<>(bus, *HttpStatus*.***OK***);

}

*@GetMapping*("/addBus")

*@ResponseBody*

public String addBus(*@RequestParam* String id) {

bus.add(id);

return "ID: " + id;

}

*@GetMapping*("/remBus")

*@ResponseBody*

public String remBus(*@RequestParam* String id) {

int i = Integer.*valueOf*(id);

bus.remove(i);

return "ID: " + id;

}

*@GetMapping*("/plane")

public ResponseEntity<Object> getPlane() {

return new ResponseEntity<>(plane, *HttpStatus*.***OK***);

}

*@GetMapping*("/addPlane")

*@ResponseBody*

public String addPlane(*@RequestParam* String id) {

plane.add(id);

return "ID: " + id;

}

*@GetMapping*("/remPlane")

*@ResponseBody*

public String remPlane(*@RequestParam* String id) {

int i = Integer.*valueOf*(id);

plane.remove(i);

return "ID: " + id;

}

*@GetMapping*("/ship")

public ResponseEntity<Object> getData() {

return new ResponseEntity<>(ships, *HttpStatus*.***OK***);

}

*@GetMapping*("/addShip")

*@ResponseBody*

public String addShip(*@RequestParam* String id) {

ships.add(id);

return "ID: " + id;

}

*@GetMapping*("/remShip")

*@ResponseBody*

public String remShip(*@RequestParam* String id) {

int i = Integer.*valueOf*(id);

ships.remove(i);

return "ID: " + id;

}

*@GetMapping*("/foos")

*@ResponseBody*

public String getFoos(*@RequestParam* String id) {

return "ID: " + id;

}

}