Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа № 1

«Простой веб-сервис на java»

Проверил: Выполнил:

асс. каф. ЭВМ ст. гр. 250503

И.Г. Скиба М.С. Патюпин

МИНСК 2024

# 1 Цель и задачи лабораторной работы

**1.1** Цель работы – освоить принципы разработки веб-сервиса с использованием стека Spring. Реализовать GET-endpoint.

**1.2** Задача:написать сервис, который обращаясь к стороннему ресурсу, выдаст факт о числе, в нескольких категориях:

-trivia - Извлекает случайный интересный факт о числе (по умолчанию, если он опущен).

-math - Предоставляет математические подробности и свойства числа.

-year - Извлекает историческую информацию или события, относящиеся к определенному году.

# 2 Результаты выполнения лабораторной работы

**2.1** Структура проекта

На рисунке 2.1 приведена схема проекта, использовалась послойная архитектура.

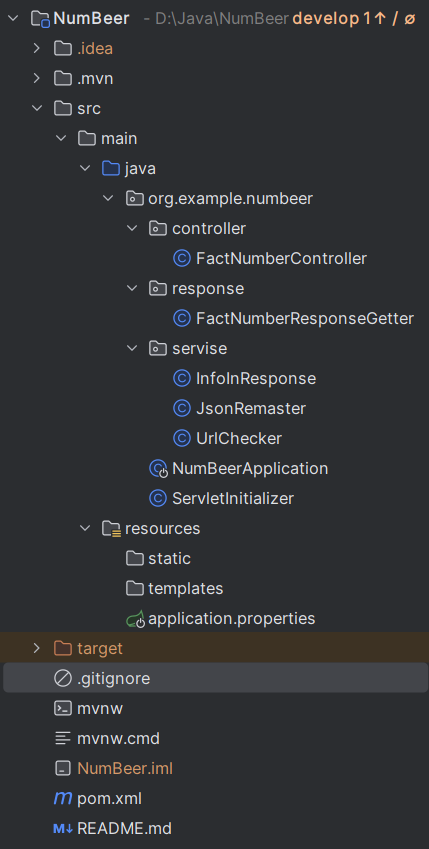


Рисунок 2.1

Папка “controller” служит для хранения классов, которые обрабатывают входящие HTTP-запросы, управляют потоком выполнения и взаимодействуют с другими компонентами приложения.

Класс “FactNumberController” является контроллером REST API и содержит два метода для обработки HTTP-запросов:

* Метод getInfo обрабатывает GET-запрос на пути /info. Он принимает два параметра: number и type, которые могут быть переданы в запросе в виде параметров. Параметры должены соответствовать регулярному выражению, указанному в аннотации @Pattern, для обеспечения безопасности. После формирует новый url, и возвращяет строку – выдержку из json ответа со стороннего сервиса.
* Метод defaultMethod обрабатывает GET-запрос на любом пути, который не соответствует /info. Он возвращает ResponseEntity с сообщением о том, что необходимо указать действительный путь.

Папка “response” служит для хранения классов, отвечающих за формирование и обработку ответов, возвращаемых веб-приложением.

Класс” FactNumberResponseGetter” отвечает за получение конечного ответа на основе указанного URL путем выполнения HTTP GET-запроса и обработки JSON-ответа.

Папка “service” служит для хранения классов, которые предоставляют бизнес-логику и функциональность, связанную с обработкой данных и выполнением операций над ними.

Класс “InfoInResponse” определяет структуру данных для хранения информации, полученной в ответе.

Класс “JsonRemaster” предоставляет метод gettingJsonKeyValues, который использует библиотеку Gson для преобразования JSON-данных в объекты Java. Метод извлекает значение поля text из JSON-ответа и возвращает его в виде строки.

Класс “UrlChecker” предоставляет метод createNewUrl, который создает и проверяет URL-адреса для получения информации о числах. Он обеспечивает контроль над разрешенными URL-адресами и проверяет код ответа HTTP.

**2.2** Листинг компьютерной программы

Файл FactNumberController

package org.example.numbeer.controller;

import jakarta.validation.constraints.Pattern;

import org.example.numbeer.response.FactNumberResponseGetter;

import org.example.numbeer.servise.UrlChecker;

import org.springframework.format.annotation.NumberFormat;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.validation.annotation.Validated;

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

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

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

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

import java.io.IOException;

@RestController

@Validated

public class FactNumberController {

@GetMapping(value = "/info")

public String getInfo(

@RequestParam(value = "number", defaultValue = "random")

@Pattern(regexp = "\\d+|^(random)")

@NumberFormat(style = NumberFormat.Style.NUMBER) String number,

@RequestParam(value = "type", defaultValue = "trivia")

@Pattern(regexp = "^(year|math|trivia)$") String type) {

String url;

try {

url = UrlChecker.createNewUrl(number, type);

} catch (IOException e) {

return "Error occurred while creating the URL: " + e.getMessage();

}

return FactNumberResponseGetter.gettingFinalResponse(url);

}

@GetMapping(value = "/\*\*")

public ResponseEntity<String> defaultMethod() {

return new ResponseEntity<>("Please specify a valid path. For example, http://localhost:8081/info?number=5&type=math", HttpStatus.BAD\_REQUEST);

}

@ExceptionHandler(IOException.class)

public ResponseEntity<Object> handleIOException() {

return new ResponseEntity<>("STATUS: 500. Internal Server Error", HttpStatus.INTERNAL\_SERVER\_ERROR);

}

}

Файл FactNumberResponseGetter

package org.example.numbeer.response;

import org.example.numbeer.servise.JsonRemaster;

import org.springframework.http.ResponseEntity;

import org.springframework.web.client.RestTemplate;

public class FactNumberResponseGetter {

private FactNumberResponseGetter() {

}

public static String gettingFinalResponse(String url) {

RestTemplate restTemplate = new RestTemplate();

ResponseEntity<String> response = restTemplate.getForEntity(url, String.class);

String jsonResponse = response.getBody();

return JsonRemaster.gettingJsonKeyValues(jsonResponse);

}

}

Файл InfoInResponse

package org.example.numbeer.servise;

public class InfoInResponse {

String text;

String number;

String found;

String type;

}

Файл JsonRemaster

package org.example.numbeer.servise;

import com.google.gson.Gson;

public class JsonRemaster {

private JsonRemaster() {

}

public static String gettingJsonKeyValues(String jsonResponse) {

Gson gson = new Gson();

InfoInResponse info;

info = gson.fromJson(jsonResponse, InfoInResponse.class);

return info.text;

}

}

Файл UrlChecker

package org.example.numbeer.servise;

import org.springframework.http.HttpStatus;

import java.io.IOException;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.ArrayList;

import java.util.List;

public class UrlChecker {

private UrlChecker() {

}

public static String createNewUrl(String number, String type) throws IOException {

String baseUrl = "http://numbersapi.com/";

List<String> allowedHosts = new ArrayList<>();

allowedHosts.add(baseUrl + number + "/trivia?json");

allowedHosts.add(baseUrl + number + "/math?json");

allowedHosts.add(baseUrl + number + "/year?json");

String newUrl = baseUrl + number + "/" + type + "?json";

URL updatedUrl = new URL(newUrl);

if (allowedHosts.contains(newUrl)) {

HttpURLConnection conn = (HttpURLConnection) updatedUrl.openConnection();

int responseCode = conn.getResponseCode();

if (responseCode == HttpStatus.OK.value()) {

return newUrl;

}

}

return "http://numbersapi.com/random/trivia?json";

}

}

Файл NumBeerApplication

package org.example.numbeer;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class NumBeerApplication {

public static void main(String[] args) {

SpringApplication.run(NumBeerApplication.class, args);

}

}

Файл SpringApplicationBuilder

package org.example.numbeer;

import org.springframework.boot.builder.SpringApplicationBuilder;

import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

public class ServletInitializer extends SpringBootServletInitializer {

@Override

protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {

return application.sources(NumBeerApplication.class);

}

}

**2.3** Результат выполнения компьютерной программы, представлен на рисунке 2.2 и рисунке 2.3.

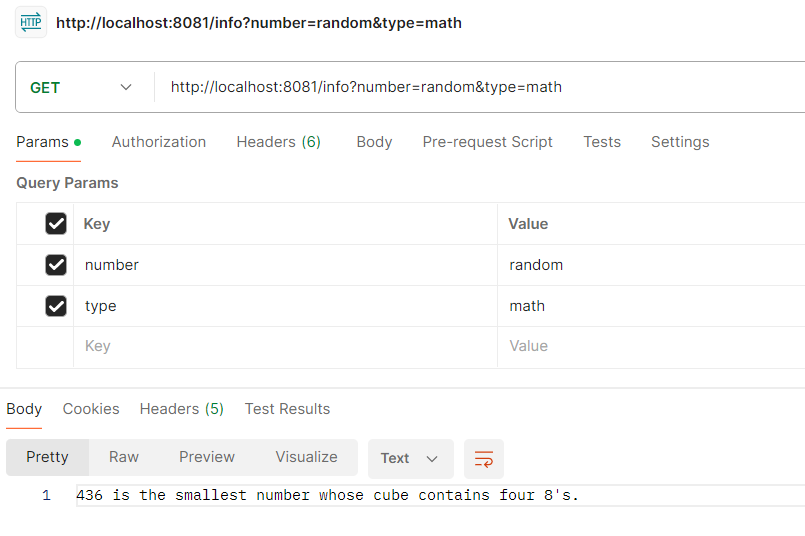


Рисунок 2.2

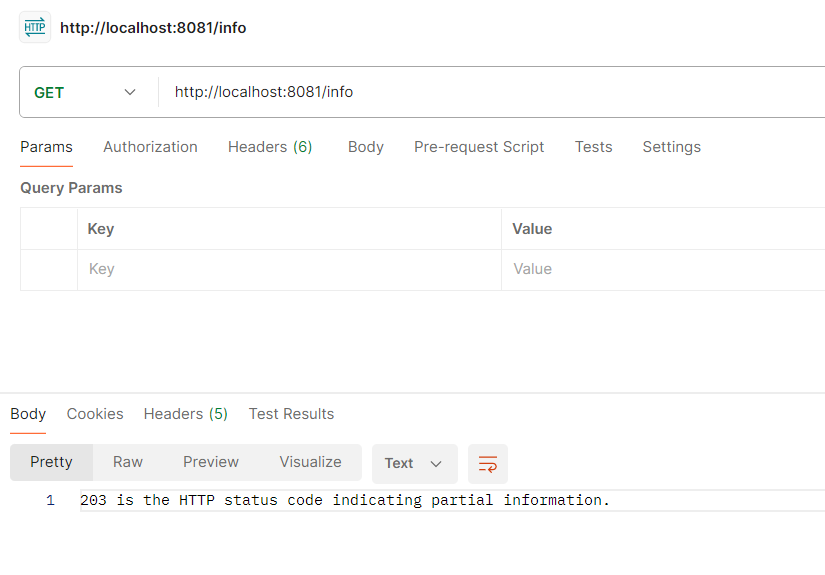


Рисунок 2.3

**2.3 Выводы по результатам выполнения ЛР**

В результате выполнения ЛР был создан простой безопасный сервис с использованием фрейморка Spring выдающий факт о числе.

Работа выполнена с использованием среды разработки Inrellj IDEA, на операционной системе Windows 10. Для тестирования API использовалось програмное обеспечение Postman. Для тестирования безопасности, и выявлении потенциально опасных фрагментов был подключен анализатор кода SonarCloud.

**3 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1 Лекционный материал по дисциплине “Программирование на языках высокого уровня”.