

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина» (УрФУ)

Институт фундаментального образования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к Лабораторной работе №2

по теме: \_\_Доработабка REST сервиса\_

по дисциплине: \_\_\_\_\_Технологии программирования\_\_\_\_\_

Студент: \_\_\_Пряминин Т. В.\_\_\_

(ФИО)

Группа:\_\_РИЗ-120938у

Преподаватель: \_\_\_\_Архипов Н. А.\_\_\_\_\_\_\_

(ФИО)

Екатеринбург

2023

**ПОСТАНОВКА ЗАДАЧИ**

1. Доработка сервиса;

2. Добавить логгирование в сервис;

3. Добавить преобразование входящего сообщения;

**РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

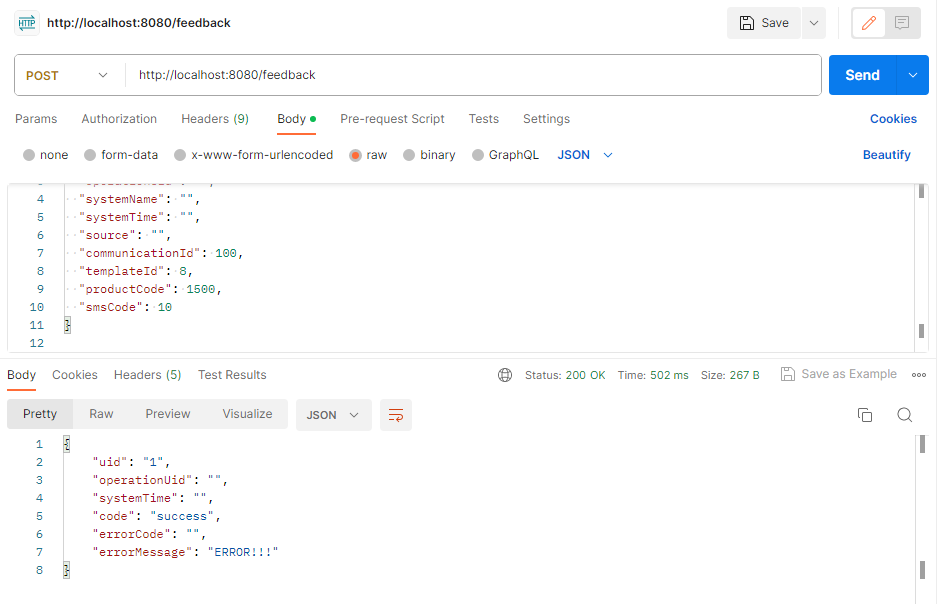


Рисунок 1 – Результат работы программы

**Оригинальный**

2023-08-13T17:29:55.398+05:00 INFO 6016 --- [nio-8080-exec-2] c.example.lab\_2.controller.MyController : ПОЛУЧЕН ЗАПРОС: Request(uid=1, operationUid=, systemName=, systemTime=, source=, communicationId=100, templateId=8, productCode=1500, smsCode=10)

**Модифицированный**

2023-08-13T17:29:55.400+05:00 INFO 6016 --- [nio-8080-exec-2] c.example.lab\_2.controller.MyController : МОДИФИЦИРОВАННЫЙ: Responce(uid=1, operationUid=, systemTime=, code=success, errorCode=, errorMessage=ERROR!!!)

**ОПИСАНИЕ ПРОГРАММЫ**

**Интерфейс MyModifyService**

public interface MyModifyService {  
 Responce modify(Responce responce);  
}

**Класс MidifyUid**

@Service  
@Qualifier("ModifyUid")  
public class ModifyUid implements MyModifyService {  
  
 @Override  
 public Responce modify(Responce responce) {  
 responce.setUid("New uid");  
 return responce;  
 }  
  
}

**Класс ModifySistemTime**

@Service  
@Qualifier("ModifySystemTime")  
public class ModifySystemTime implements MyModifyService {  
  
 @Override  
 public Responce modify(Responce responce) {  
 responce.setSystemTime("123");  
 return responce;  
 }  
  
}

**Класс ModifyErrorMessage**

@Service  
@Qualifier("ModifyErrorMessage")  
public class ModifyErrorMessage implements MyModifyService {  
  
 @Override  
 public Responce modify(Responce responce) {  
 responce.setErrorMessage("ERROR!!!");  
 return responce;  
 }  
  
}

**Класс MyController**

@Slf4j  
@RestController  
public class MyController {  
 private final MyModifyService myModifyService;  
  
 @Autowired  
 public MyController(@Qualifier("ModifyErrorMessage") MyModifyService myModifyService) {  
 this.myModifyService = myModifyService;  
 }  
  
 @PostMapping(value = "/feedback")  
 public ResponseEntity<Responce> feedback(@RequestBody Request request) {  
 *log*.info("ПОЛУЧЕН ЗАПРОС: " + String.*valueOf*(request));  
  
 Responce response = Responce.*builder*()  
 .uid(request.getUid())  
 .operationUid(request.getOperationUid())  
 .systemTime(request.getSystemName())  
 .code("success")  
 .errorCode("")  
 .errorMessage("")  
 .build();  
  
 Responce modifyedResponce = myModifyService.modify(response);  
 *log*.info("МОДИФИЦИРОВАННЫЙ: " + String.*valueOf*(response));  
  
 return new ResponseEntity<>(response, HttpStatus.*OK*);  
 }  
}

**ВЫВОД**

В ходе выполнения данной лабораторной работы, был доработан простой REST сервер. Добавлено логирование и модификация ответов.

Ссылка на github с выполненными заданиями: *https://github.com/Prodiks/spring\_labs*