

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 №4

по теме: \_\_Разработка REST full сервиса\_

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

Студент: \_\_\_Пряминин Т. В.\_\_\_

(ФИО)

Группа:\_\_РИЗ-120938у

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

(ФИО)

Екатеринбург

2023

**ПОСТАНОВКА ЗАДАЧИ**

Создать RESET full проект с СУБД H2. Выполнить указанные запросы к сервису посредством Postman.

**РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

Выборка всех студентов

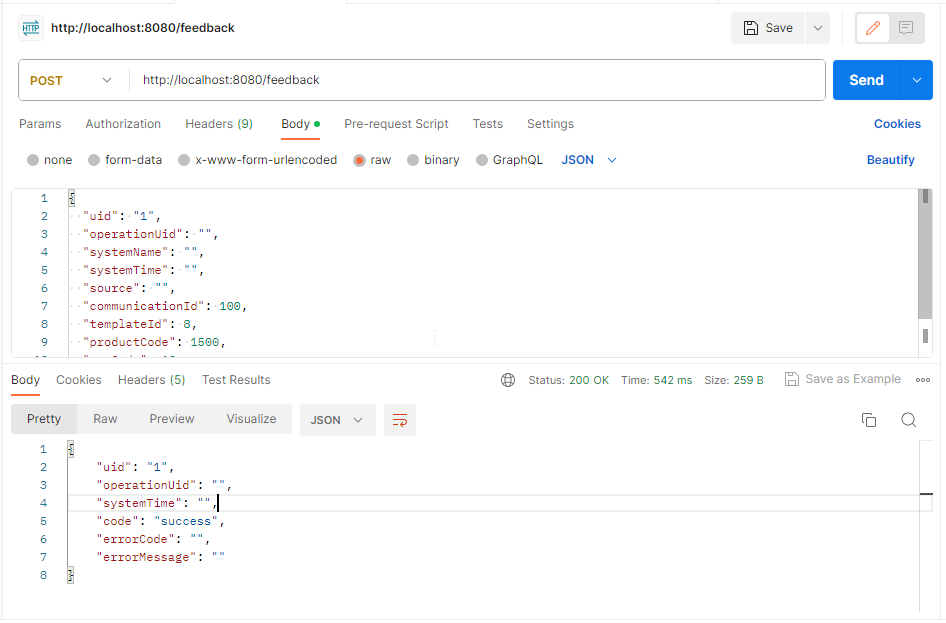


Рисунок 1 – Результат работы

Студент с id = 2

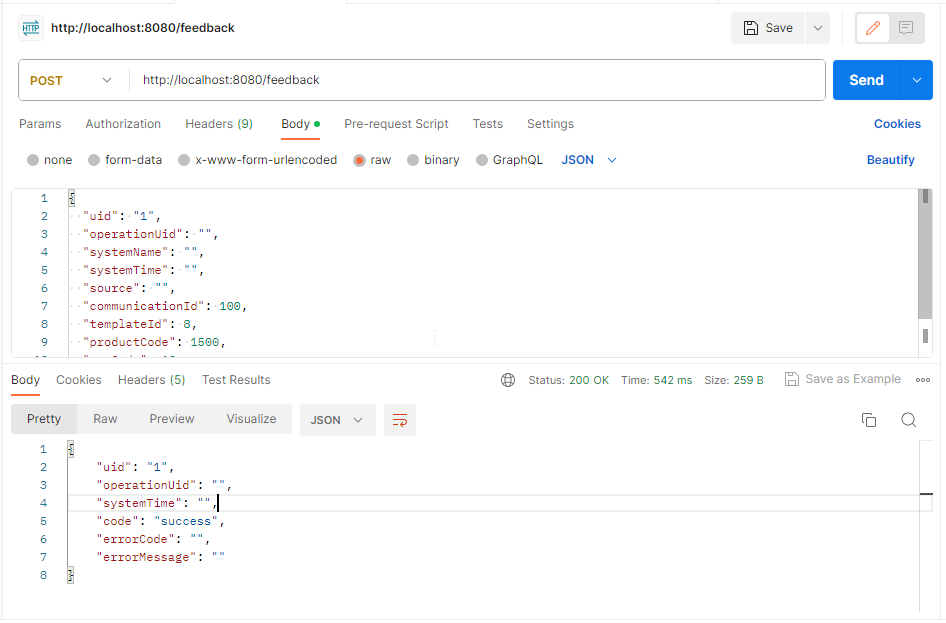


Рисунок 2 – Результат работы

**ОПИСАНИЕ ПРОГРАММЫ**

**Интерфейс MyController**

@RestController  
@RequestMapping("/api")  
public class MyController {  
 @Autowired  
 private StudentService studentService;  
  
 @GetMapping("/students")  
 public List<Student> showAllStudents() {  
 List<Student> allStudents = studentService.getAllStudents();  
 return allStudents;  
 }  
  
 @GetMapping("/students/{id}")  
 public Student getStudent(@PathVariable("id") int id) {  
 return studentService.getStudent(id);  
 }  
  
 @PostMapping("students/")  
 public Student saveStudent(@RequestBody Student student) {  
 return studentService.saveStudent(student);  
 }  
  
 @PutMapping("/students")  
 public Student updateStudent(@RequestBody Student student) {  
 return studentService.saveStudent(student);  
 }  
  
 @DeleteMapping("students/{id}")  
 public void updateStudent(@PathVariable("id") int id) {  
 studentService.deleteStudent(id);  
 }  
}

**Интерфейс StudentDAO**

@Repository  
public interface StudentDAO {  
 List<Student> getAllStudents();  
  
 Student saveStudent(Student student);  
  
 Student getStudent(int id);  
  
 void deleteStudent(int id);  
}

**Класс StudentDAOImpl**

@Slf4j  
@Repository  
public class StudentDAOImpl implements StudentDAO {  
 @Autowired  
 private EntityManager entityManager;  
  
 @Override  
 public List<Student> getAllStudents() {  
 Query query = entityManager.createQuery("from Student");  
 List<Student> allStudents = query.getResultList();  
 *log*.info("getAllStudents" + allStudents);  
 return allStudents;  
 }  
  
 @Override  
 public Student saveStudent(Student student) {  
 return entityManager.merge(student);  
  
 }  
  
 @Override  
 public Student getStudent(int id) {  
 return entityManager.find(Student.class, id);  
 }  
  
 @Override  
 public void deleteStudent(int id) {  
 Query query = entityManager.createQuery("delete from Student " + " where id = :studentId");  
 query.setParameter("studentId", id);  
 query.executeUpdate();  
 }  
}

**Интерфейс Student**

@Getter  
@Setter  
@Entity  
@Table(name = "STUDENTS")  
public class Student {  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 @Column(name = "id")  
 private int id;  
  
 @Column(name = "name")  
 private String name;  
  
 @Column(name = "surname")  
 private String surname;  
  
 @Column(name = "faculty")  
 private String faculty;  
  
 @Column (name = "age")  
 private int age;  
  
 public Student() {  
 }  
  
 public Student(String name, String surname, String faculty, int age) {  
 this.name = name;  
 this.surname = surname;  
 this.faculty = faculty;  
 this.age = age;  
 }  
}

**Интейрфейс StudentService**

@Service  
public interface StudentService {  
 List<Student> getAllStudents();  
 Student getStudent(int id);  
 Student saveStudent(Student student);  
 void deleteStudent(int id);  
}

**Класс** **StudentServiceImpl**

@Service  
public class StudentServiceImpl implements StudentService {  
 @Autowired  
 private StudentDAO studentDAO;  
  
 @Override  
 @Transactional  
 public List<Student> getAllStudents() {  
 return studentDAO.getAllStudents();  
 }  
  
 @Override  
 @Transactional  
 public Student getStudent(int id) {  
 return studentDAO.getStudent(id);  
 }  
  
 @Override  
 @Transactional  
 public Student saveStudent(Student student) {  
 return studentDAO.saveStudent(student);  
 }  
  
 @Override  
 @Transactional  
 public void deleteStudent(int id) {  
 studentDAO.deleteStudent(id);  
 }  
}

**ВЫВОД**

В ходе выполнения данной лабораторной работы, были получены навыки работы с СУБД H2 и создан простой REST full сервис.

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