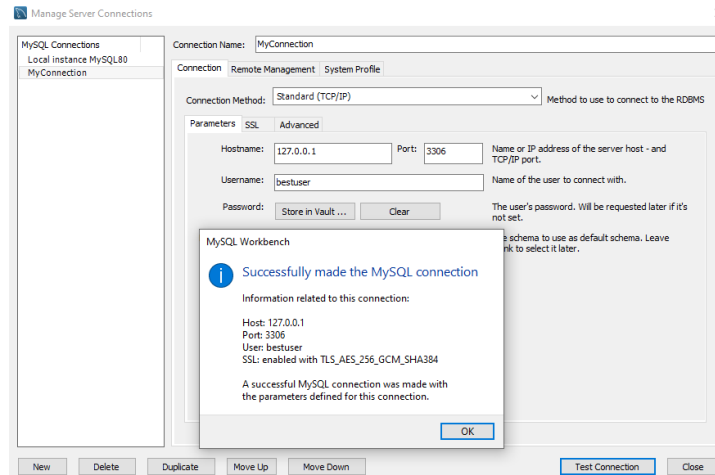


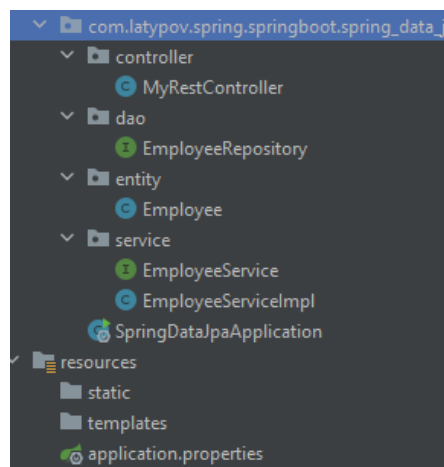
24. Spring Framework (DATA JPA)

- Модификация БД посредством интерфейса JpaRepository

Конфигурация подключения в среде MySQL:



Структура проекта:



application.properties (параметры подключения приложения к БД):

```
spring.datasource.url=jdbc:mysql://localhost:3306/my_db?useSSL=false&serverTimezone=UTC
spring.datasource.username=bestuser
spring.datasource.password=bestuser
server.port=3333
```

Создание таблицы Employees:

```
CREATE TABLE my_db.employees (
  id int NOT NULL AUTO_INCREMENT,
  name varchar(15),
  surname varchar(25),
  department varchar(20),
  salary int,
  PRIMARY KEY (id);
```

entity.Employee (связь с таблицей Employees):

```
@Entity
@Table(name = "employees")
public class Employee {
  @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 = "department")
  private String department;
  @Column(name = "salary")
  private int salary;
```

dao.EmployeeRepository (интерфейс доступа к данным, наследование от класса JpaRepository):

```
public interface EmployeeRepository extends JpaRepository<Employee,Integer> {
  public List<Employee> findAllByName(String name);
}
```

service.EmployeeService (сервисная часть, интерфейс с методами):

```
public interface EmployeeService {
  public List<Employee> getAllEmployees();
  public void saveEmployee(Employee employee);
  public Employee getEmployee(int id);
  public void deleteEmployee(int id);
  public List<Employee> findAllByName(String name);
}
```

service.EmployeeServiceImpl (имплементация методов, встроенных в библиотеку JpaRepository findAll(), save(), findById(), get(), deleteById(), findAllByName(), реализация методов не требуется):

```
@Service
public class EmployeeServiceImpl implements EmployeeService {
```

```

@Autowired
private EmployeeRepository employeeRepository;

@Override
//без аннотации Transactional
public List<Employee> getAllEmployees() {
    return employeeRepository.findAll();
}

@Override
public void saveEmployee(Employee employee) {
    employeeRepository.save(employee);
}

@Override
public Employee getEmployee(int id) {
    Employee employee = null;
    Optional<Employee> optional = employeeRepository.findById(id);
    if (optional.isPresent()) {
        employee = optional.get();
    }
    return employee;
}

@Override
public void deleteEmployee(int id) {
    employeeRepository.deleteById(id);
}

@Override
public List<Employee> findAllByName(String name) {
    List<Employee> emps = employeeRepository.findAllByName(name);
    return emps;
}
}

```

controller.MyRestController (реализация контроллера):

```

@RestController
@RequestMapping("/api")
public class MyRestController {
    @Autowired
    private EmployeeService employeeService;

    @GetMapping("/employees")
    public List<Employee> showAllEmployees() {
        List<Employee> allEmployees = employeeService.getAllEmployees();
        return allEmployees;
    }

    @GetMapping("/employees/{id}")
    public Employee getEmployee(@PathVariable int id) {
        Employee employee = employeeService.getEmployee(id);

        return employee;
    }

    @PostMapping("/employees")
    public Employee addNewEmployee(@RequestBody Employee employee) {

```

```

        employeeService.saveEmployee(employee);
        return employee;
    }

    @PutMapping("/employees")
    public Employee updateEmployee(@RequestBody Employee employee) {
        employeeService.saveEmployee(employee);
        return employee;
    }

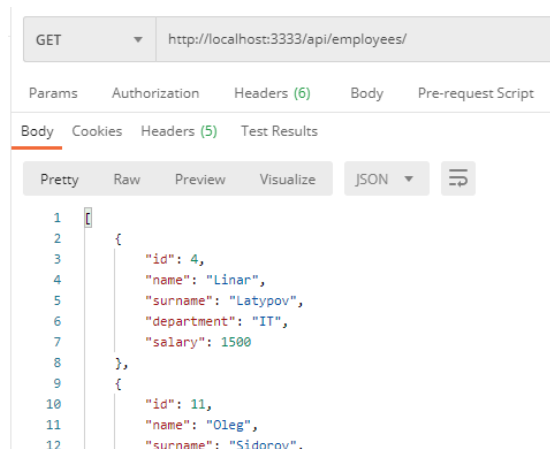
    @DeleteMapping("/employees/{id}")
    public String deleteEmployee(@PathVariable int id) {
        Employee employee = employeeService.getEmployee(id);
        employeeService.deleteEmployee(id);
        return "Employee with ID= " + id + " was deleted";
    }

    @GetMapping("/employees/name/{name}")
    public List<Employee> showAllEmpsByName(@PathVariable String name) {
        List<Employee> employees = employeeService.findAllByName(name);
        return employees;
    }
}

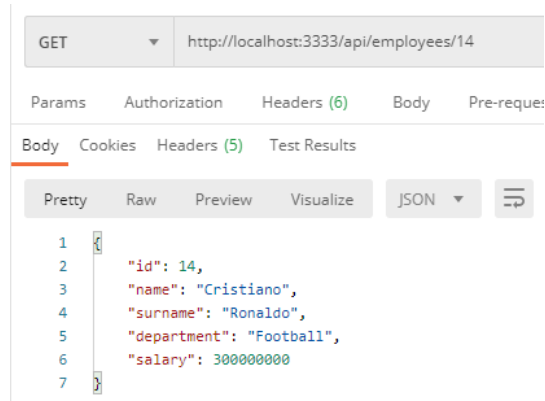
```

Запуск приложения и отправка запросов через Postman

Get-запрос:



GetById-запрос:



GetByName-запрос:

GET <http://localhost:3333/api/employees/name/Cristiano>

Params Authorization Headers (6) Body Pre-request Script Tests S

Query Params

KEY	VALUE
-----	-------

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 14,
4     "name": "Cristiano",
5     "surname": "Ronaldo",
6     "department": "Football",
7     "salary": 30000000
8   },
9   {
10    "id": 15,
11    "name": "Cristiano",
12    "surname": "Ronaldo",
13    "department": "Football",
14    "salary": 15000000
15  }
16 ]
```

Post-запрос:

POST <http://localhost:3333/api/employees>

Params Authorization Headers (8) Body Pre-request Script

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw

```
1 {
2   "name": "Cristiano",
3   "surname": "Ronaldo",
4   "department": "Football",
5   "salary": 15000000
6 }
```

Put-запрос:

PUT <http://localhost:3333/api/employees>

Params Authorization Headers (8) Body Pre-request Script

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary

```
1 {
2   "id": 14,
3   "name": "Cristiano",
4   "surname": "Ronaldo",
5   "department": "Football",
6   "salary": 30000000
7 }
```

Delete-запрос:

DELETE ▼

http://localhost:3333/api/employees/14

Params

Authorization

Headers (6)

Body

Pre-request Scri

Body

Cookies

Headers (5)

Test Results


Pretty

Raw

Preview

Visualize

Text ▼



1

Employee with ID= 14 was deleted