

SpringBoot练习—苍穹外卖为例

1.前后端联调

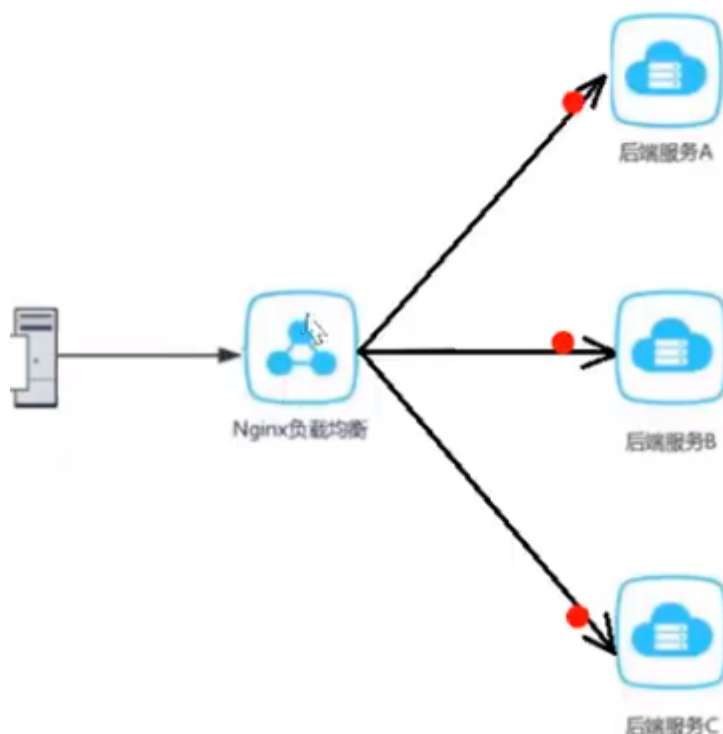
nginx反向代理，将前端发送的动态请求由nginx转发到后端服务器



1.1 nginx反向代理的好处:

1.提高访问速度：可以在nginx层做缓存，我们如果请求同样的接口地址，就无需再请求后端接口，直接在nginx这把缓存数据响应给前端

2.进行负载均衡



3.保证后端服务的安全，后端部署在公司内网，不对外开发，外部请求到达nginx服务器，在转发到内网的后端进行响应。

1.2nginx反向代理的配置方式

```

server{
    listen 80;
    server_name localhost;

    location /api/ {
        proxy_pass http://localhost:8080/admin/; #反向代理
    }
}

```

nginx.conf



http://localhost/api/employee/login

NGINX

http://localhost:8080/admin/employee/login

1.3nginx负载均衡的配置方式

nginx 负载均衡的配置方式：

```

upstream webservers{
    server 192.168.100.128:8080;
    server 192.168.100.129:8080;
}

server{
    listen 80;
    server_name localhost;

    location /api/ {
        proxy_pass http://webservers/admin/; #负载均衡
    }
}

```

nginx.conf

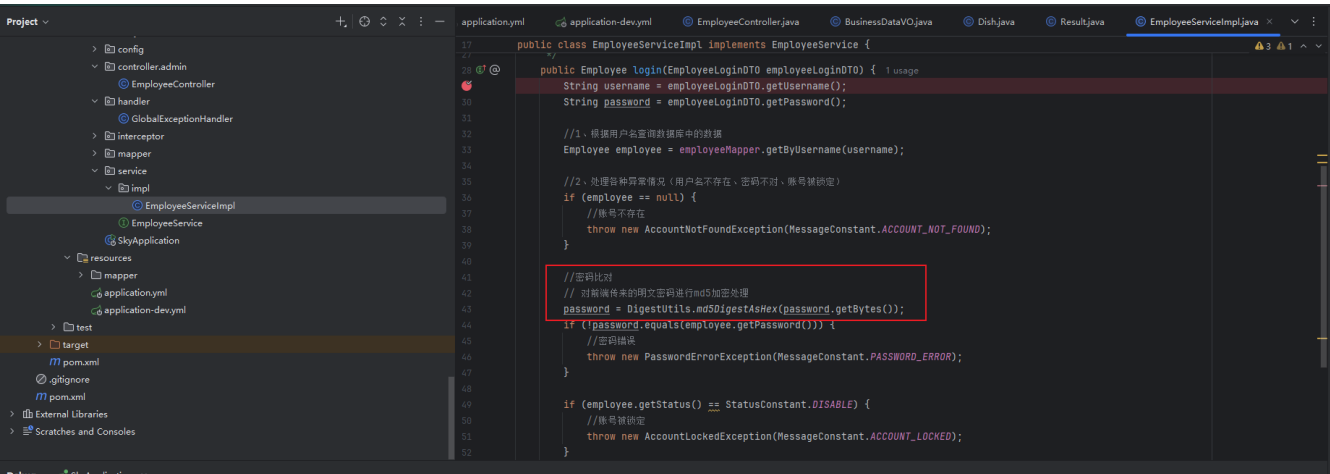
负载均衡策略：

nginx 负载均衡策略：

名称	说明
轮询	默认方式
weight	权重方式，默认为1，权重越高，被分配的客户端请求就越多
ip_hash	依据ip分配方式，这样每个访客可以固定访问一个后端服务
least_conn	依据最少连接方式，把请求优先分配给连接数少的后端服务
url_hash	依据url分配方式，这样相同的url会被分配到同一个后端服务
fair	依据响应时间方式，响应时间短的服务将会被优先分配

2.功能点一：完善登录功能，对密码进行MD5加密

数据库存储的密码存储的是md5加密之后的，在EmployeeServiceImpl中，对获取到的前端输入密码进行md5加密，然后再进行和数据库比较



```
17 public class EmployeeServiceImpl implements EmployeeService {
18     // ...
19     public Employee login(EmployeeLoginDTO employeeLoginDTO) {
20         // 1. 根据用户名查询数据库中的数据
21         String username = employeeLoginDTO.getUsername();
22         Employee employee = employeeMapper.getByUsername(username);
23         // 2. 处理各种异常情况（用户名不存在、密码不对、账号被锁定）
24         if (employee == null) {
25             // 账号不存在
26             throw new AccountNotFoundException(MessageConstant.ACCOUNT_NOT_FOUND);
27         }
28         // 密码比对
29         // 对前端传来的明文密码进行md5加密处理
30         password = DigestUtils.md5DigestAsHex(password.getBytes());
31         if (!password.equals(employee.getPassword())) {
32             // 密码错误
33             throw new PasswordErrorException(MessageConstant.PASSWORD_ERROR);
34         }
35         if (employee.getStatus() == StatusConstant.DISABLE) {
36             // 账号被锁定
37             throw new AccountLockedException(MessageConstant.ACCOUNT_LOCKED);
38         }
39         return employee;
40     }
41 }
```

3.导入接口文档

apifox:
ifox: 接口管理->根目录->三点->添加子目录->命名名字->三点->导入->选择yapi->拖入文件

4.Swagger

使用Swagger你只需要按照它的规范去定义接口，及接口相关信息，就可以生成接口文档，以及在线调试页面。
Knife4j是为Java MVC框架集成Swagger生成Api文档的增强解决方案。

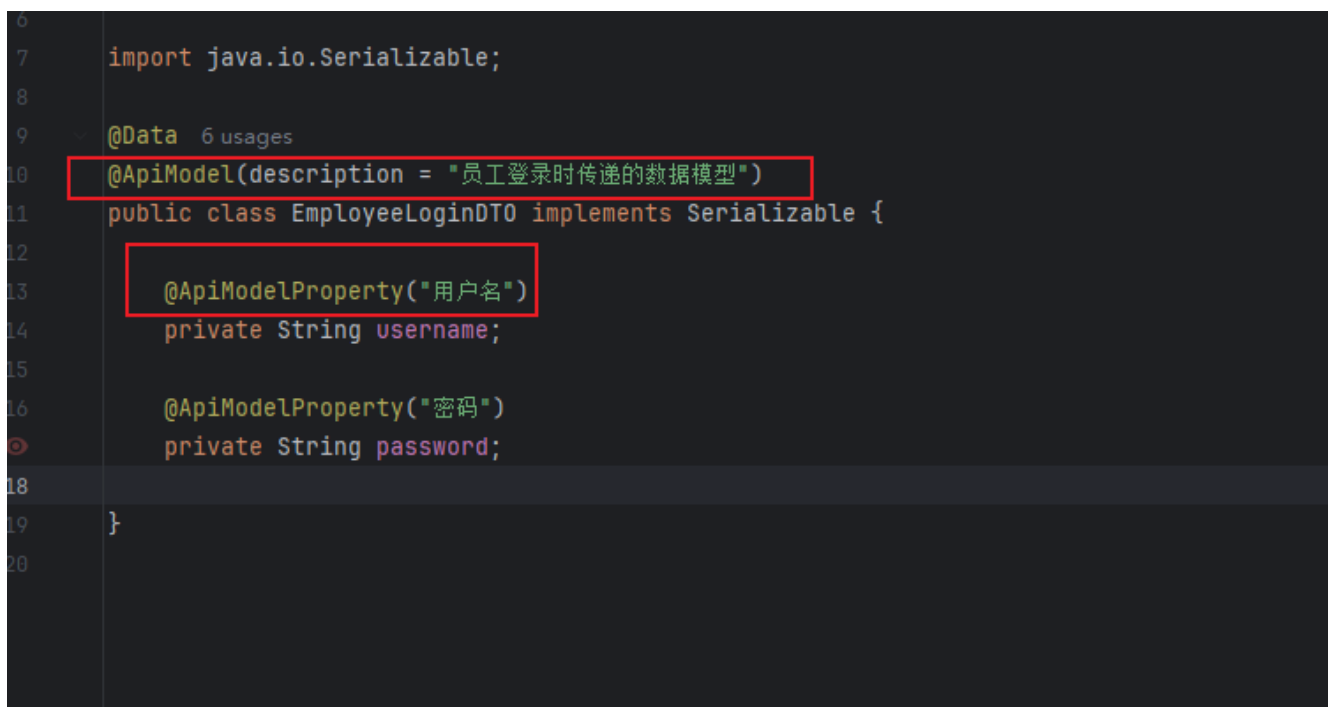
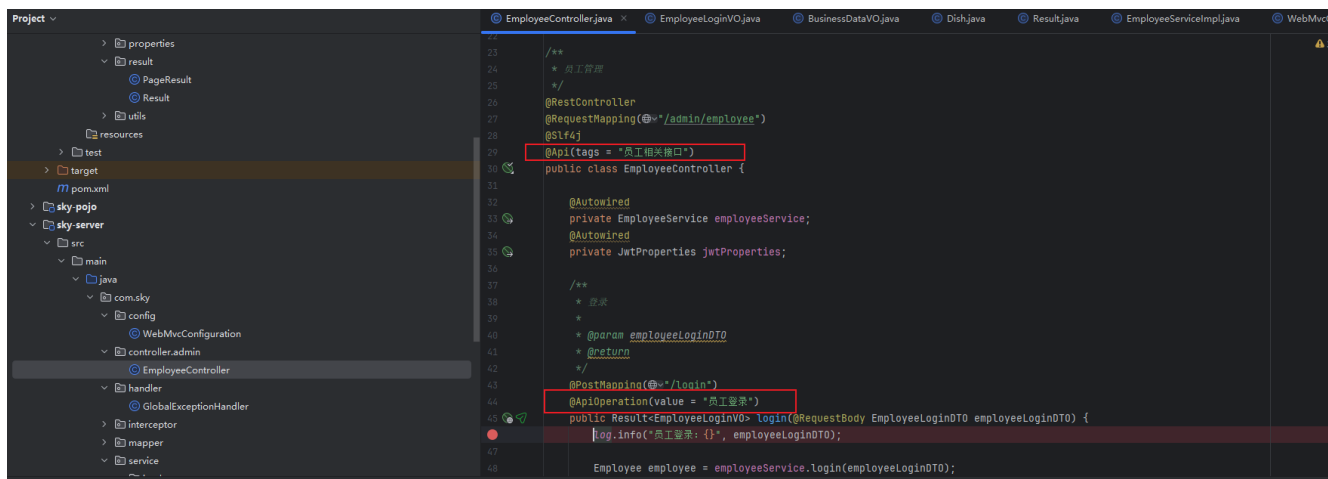
辨析Swagger和Yapi:

Yapi是设计阶段使用的工具，管理和维护接口。

Swagger是开发阶段使用的框架，帮助后端开发人员做后端的接口测试。

4.1Swagger常见注解:

注解	说明
@Api	用在类上，例如Controller，表示对类的说明
@ApiModel	用在类上，例如Entity、DTO、VO
@ApiModelProperty	用在属性上，描述属性信息
@ApiOperation	用在方法上，例如Controller的方法，说明方法的用途、作用



5 返回值封装

主要为三部分：code,data,msg

code就是状态码，不同的数字代表不同的状态，前端要根据后端的code做出对应的提示信息，这一项必须要有

data,就是返回数据，一般在查询操作时会用到，把后端查询到的数据保存到data响应给前端

msg就是文字信息，返回给前端一些提示

6.什么时候使用DTO，什么时候使用Entity

以登录举例，登录前端提交来的数据，只是实体类对应的属性的一部分，也就是他们差别较大时，建议使用DTO来封装数据

根据新增员工接口设计对应的DTO：

Body

名称	类型	是否必须	默认值	备注	其他信息
id	integer	非必须		员工id	format: int64
idNumber	string	必须		身份证	
name	string	必须		姓名	
phone	string	必须		手机号	
sex	string	必须		性别	
username	string	必须		用户名	

```
private Long id;

private String username;

private String name;

private String password;

private String phone;

private String sex;

private String idNumber;

private Integer status;

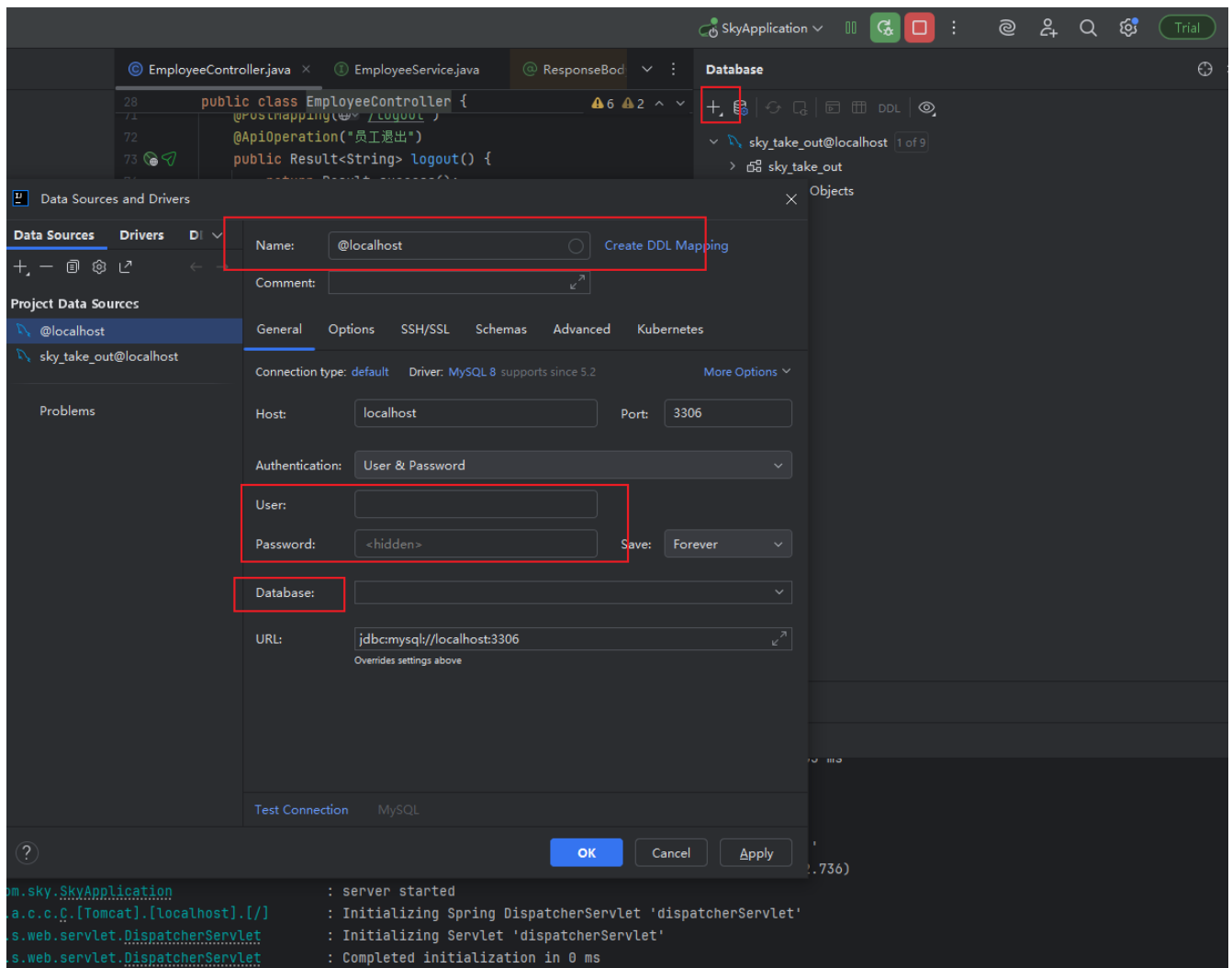
private LocalDateTime createTime;

private LocalDateTime updateTime;

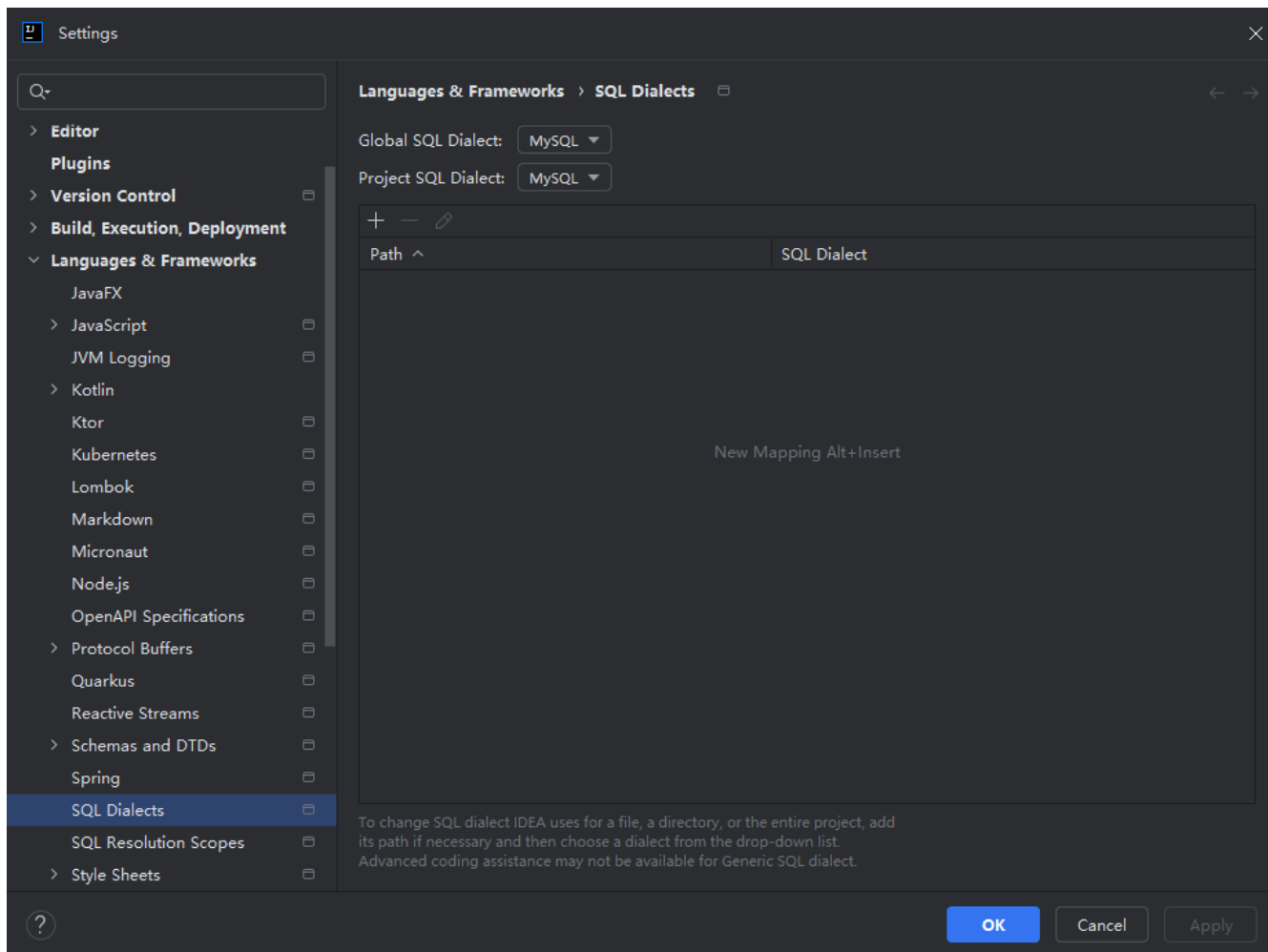
private Long createUser;

private Long updateUser;
```

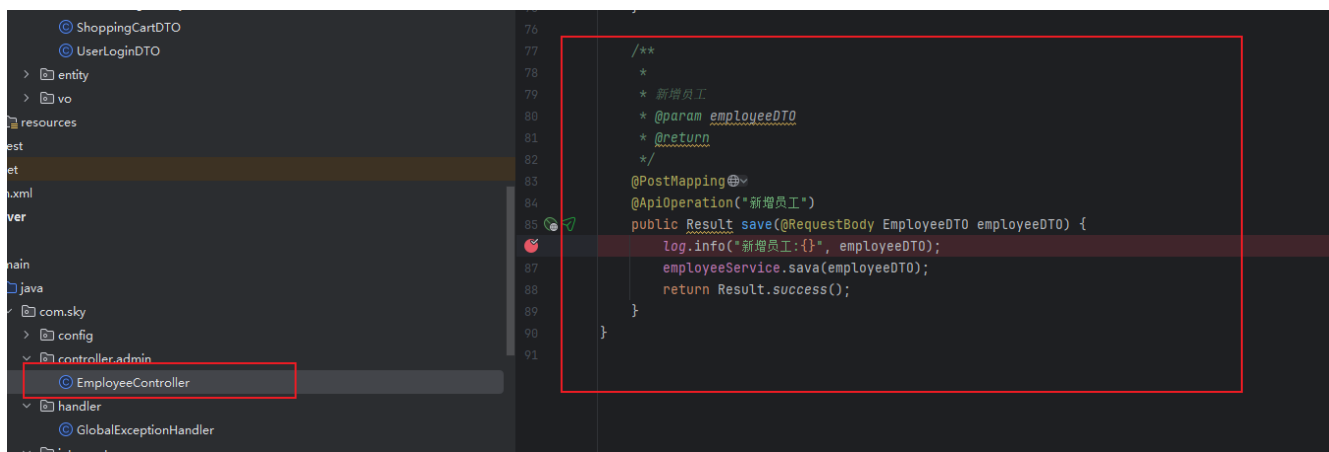
8.idea连接数据库设置细节

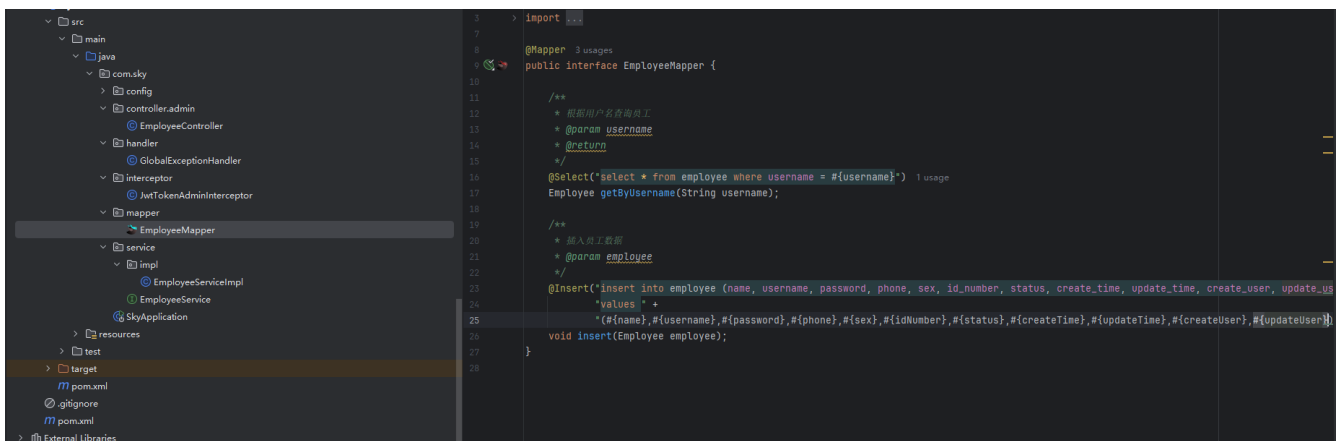
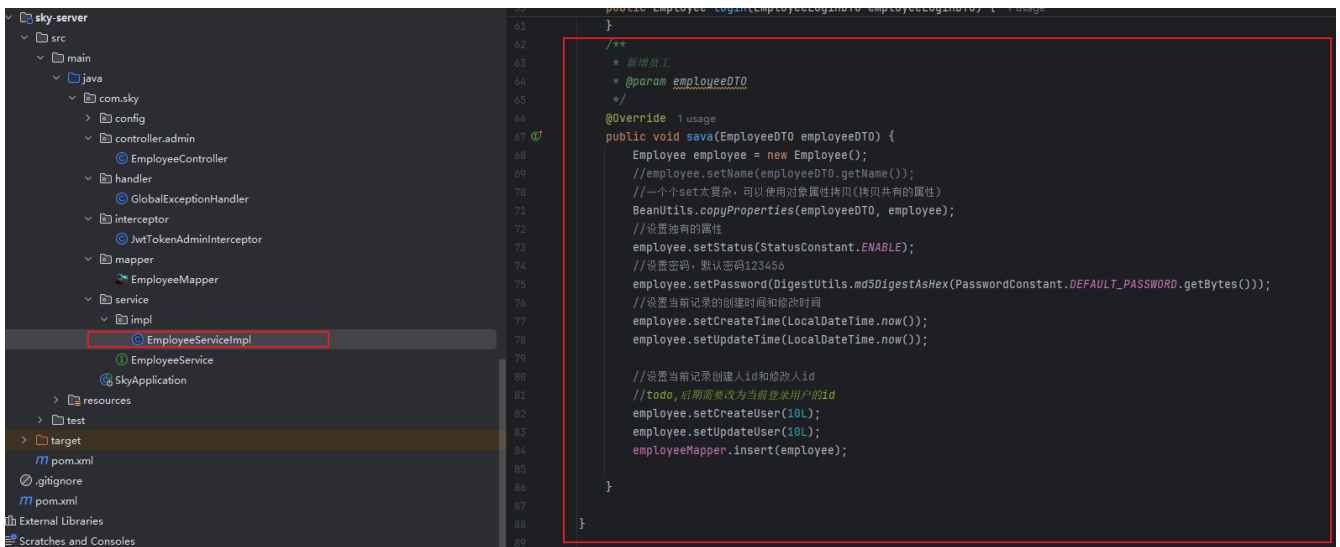
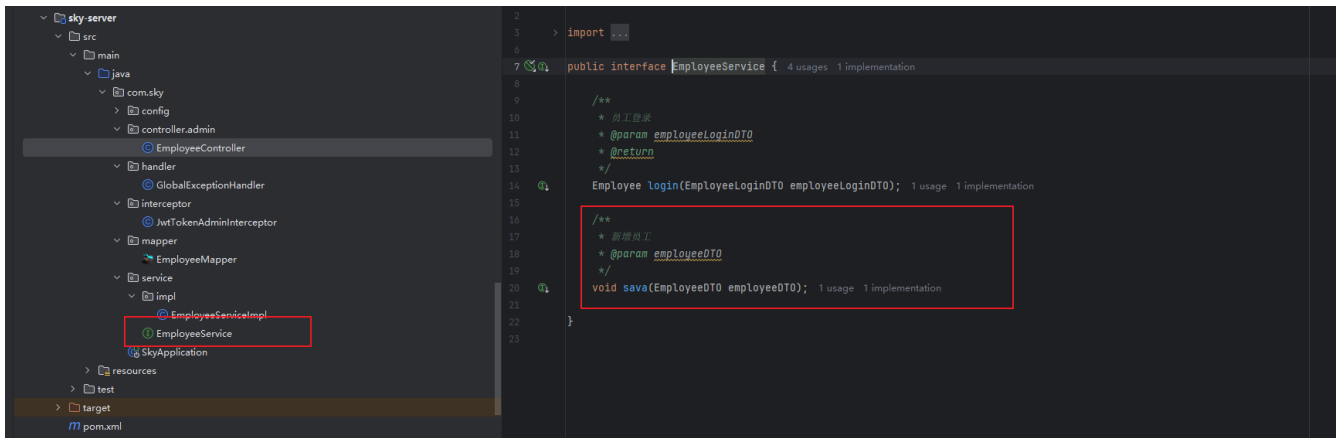


设置完这个之后，要打开setting,把Global SQL Dialect和Project SQL Dialect设置为MySQL



9功能点二：新增员工功能实现





使用swagger一开始报401

原因：因为项目有jwt,没有token不能执行成功

在swagger的文档管理模块下添加全局参数设置，设置token，token的值通过接口调试登录功能的返回值里获取

default

🏠 主页

🔗 Swagger Models

📁 文档管理

⚙️ 全局参数设置

📄 离线文档

⚙️ 个性化设置

👤 员工相关接口

苍穹外卖项目接口文档

输入文档关键字搜索 🔍 中

主页 全局参数设置(default) X 新增员工 X

Knife4j 提供全局参数Debug功能,目前默认提供header(请求头)、query(form)两种方式的入参
在此添加全局参数后,默认Debug调试tab页会带上该参数

+ 添加参数

参数名称	参数值	参数类型	操作
token	eyJhbGciOiJIUzI1NiU9IjEyJm50IjE6IjEwIjEzX2hwIjoiYXNzYyMTY3MDC2IjQISTNaNGhqbQzwc2I5uG4JwknP4SCpH4OSZ1Mkr4JzK	header ✓	删除

🏠 主页

🔗 Swagger Models

📁 文档管理

👤 员工相关接口

POST 新增员工

POST 员工登录

POST 员工退出

主页 全局参数设置(default) X 新增员工 X

📄 文档

★ 调试

🔗 Open

POST /admin/employee 发送

🔴 请求头部 请求参数 AfterScript

☐ x-www-form-urlencoded ☐ form-data ☒ raw JSON(application/json) v

```
1- {
2
3   "idNumber": "111111222377776666",
4   "name": "张三",
5   "phone": "12313217654",
6   "sex": "1",
7   "username": "zhangsai"
8 }
```

📄 响应内容 Raw Headers Curl

显示说明 响应码: 200 耗时: 4.0s 大小: 33 B

```
1- {
2   "code": 1,
3   "msg": null,
4   "data": null
5 }
```