

SpringBoot练习—苍穹外卖为例

1.前后端联调

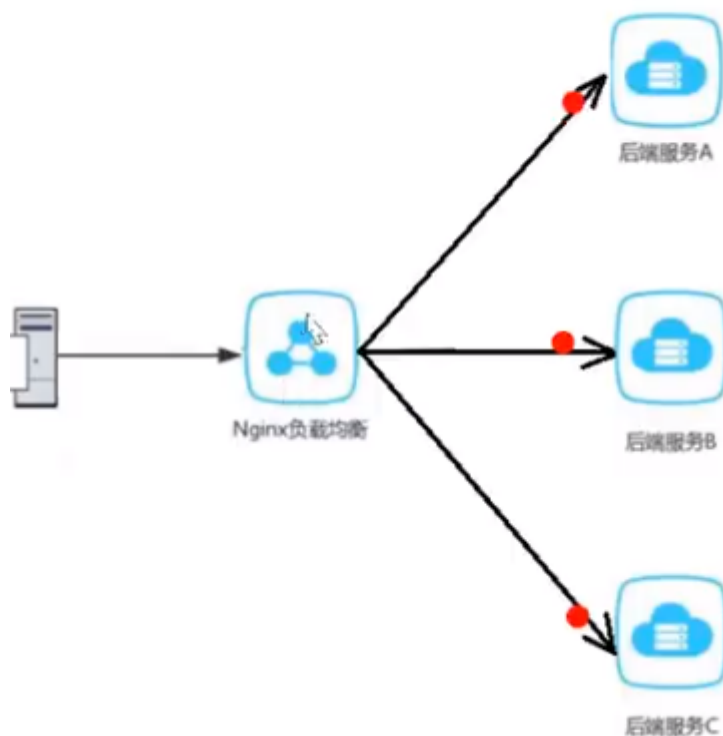
nginx反向代理，将前端发送的动态请求由nginx转发到后端服务器



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.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

2.1Swagger

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。

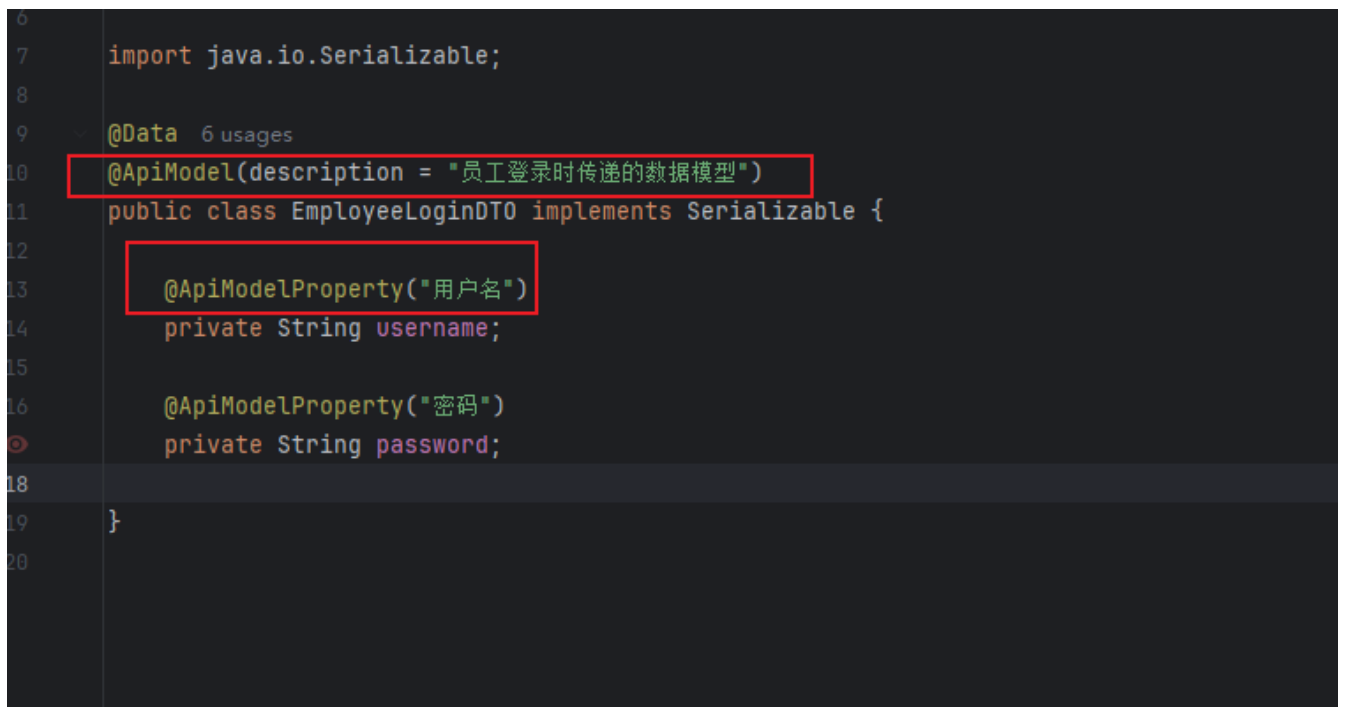
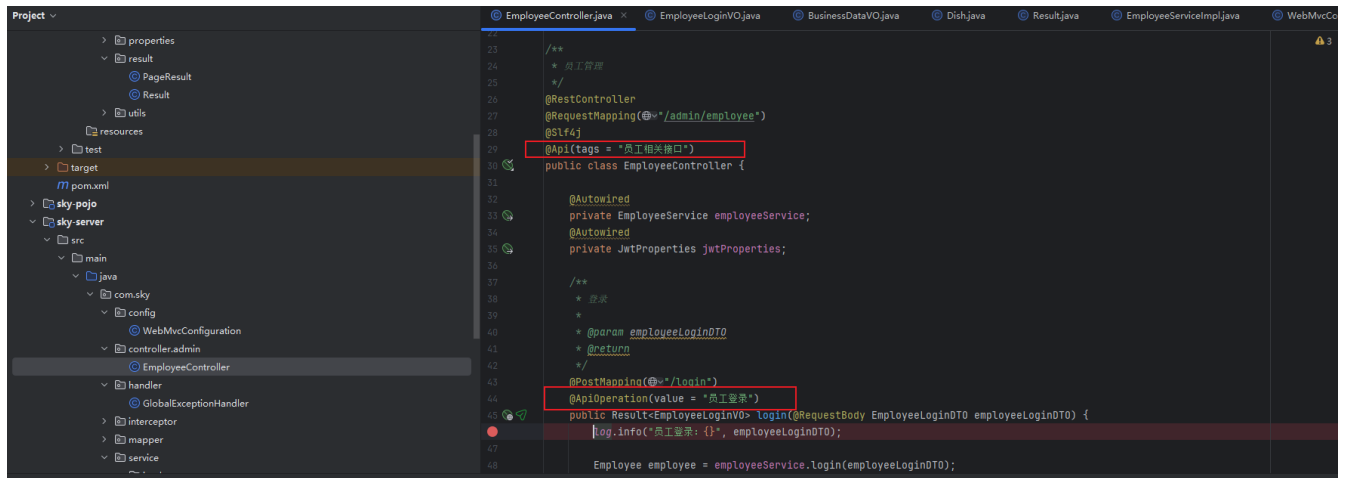
辨析Swagger和Yapi:

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。

2.1.1Swagger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



3.其他准备

3.1.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

3.2.什么时候使用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;

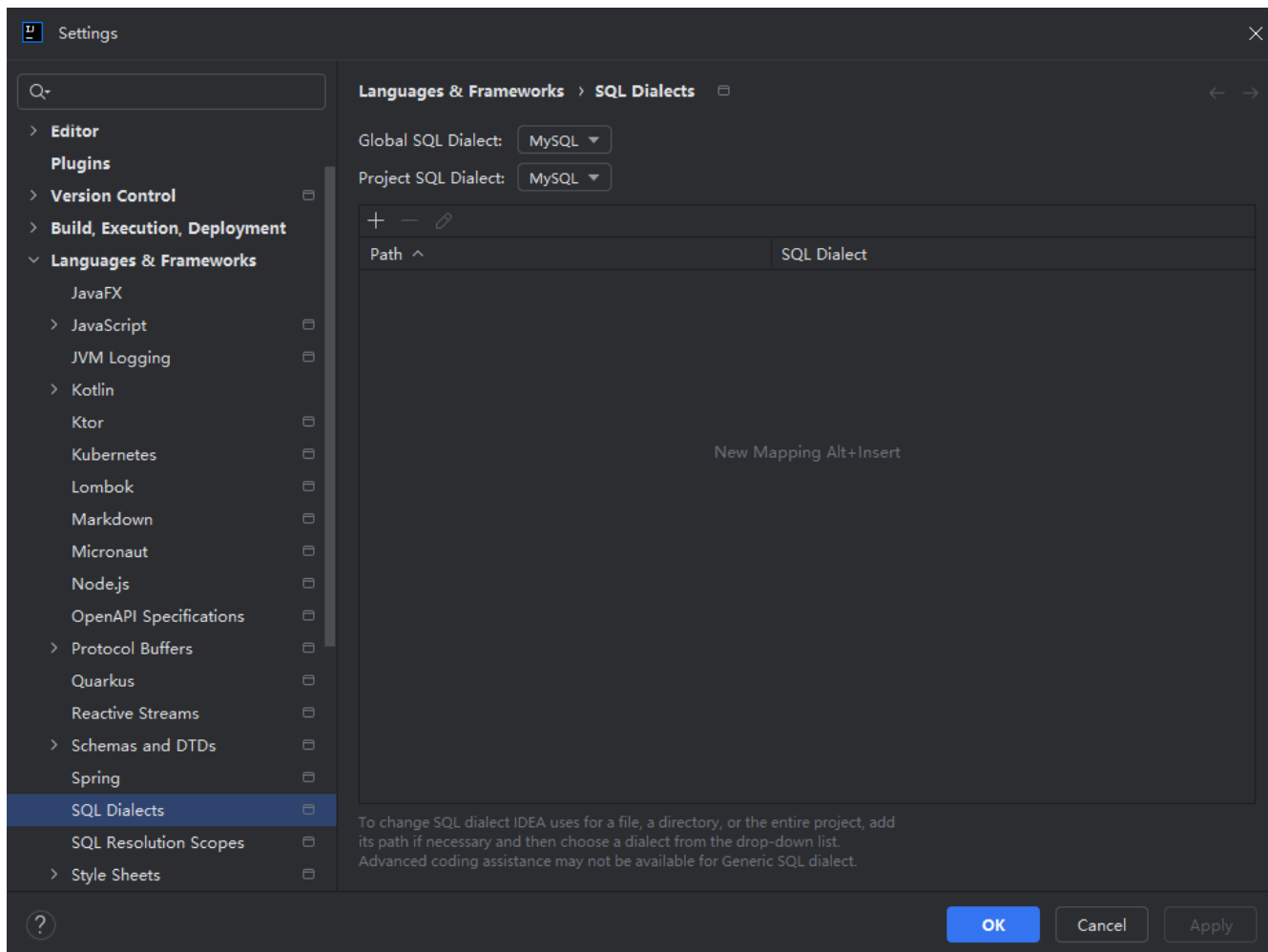
private Long createUser;

private Long updateUser;
```

3.3.idea连接数据库设置细节

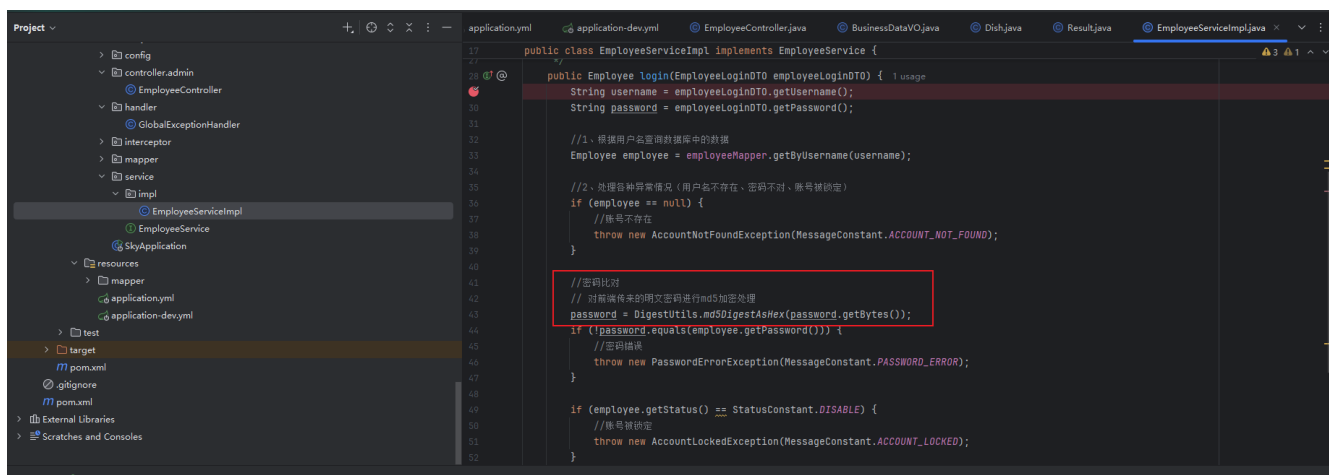
The screenshot shows the IntelliJ IDEA interface with the 'Data Sources and Drivers' window open. The window is divided into several tabs: 'Data Sources', 'Drivers', 'General', 'Options', 'SSH/SSL', 'Schemas', 'Advanced', and 'Kubernetes'. The 'Data Sources' tab is active, showing a list of data sources on the left. The 'General' tab is selected, displaying the configuration for the '@localhost' data source. The configuration includes fields for Name, Comment, Host, Port, Authentication, User, Password, Database, and URL. The 'Test Connection' button is highlighted. The background shows the code editor with the EmployeeController.java file open, and the bottom status bar indicates the server is started.

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

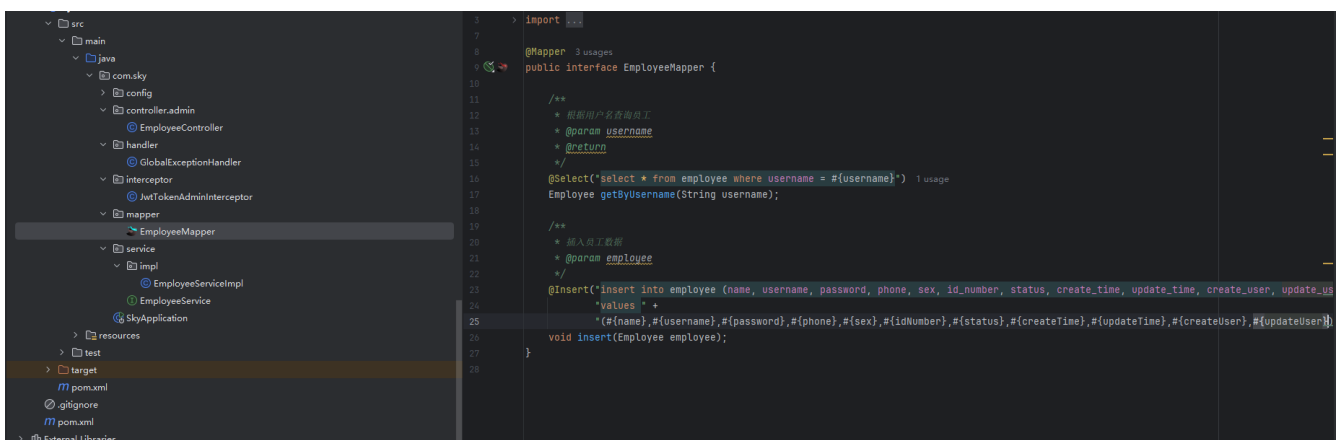
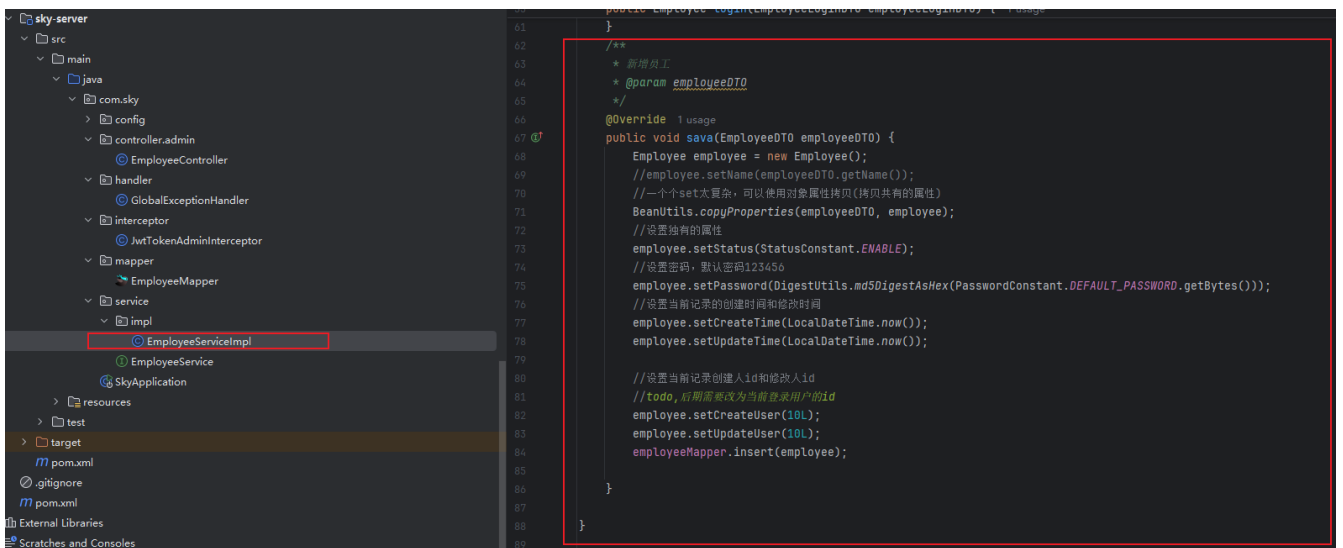
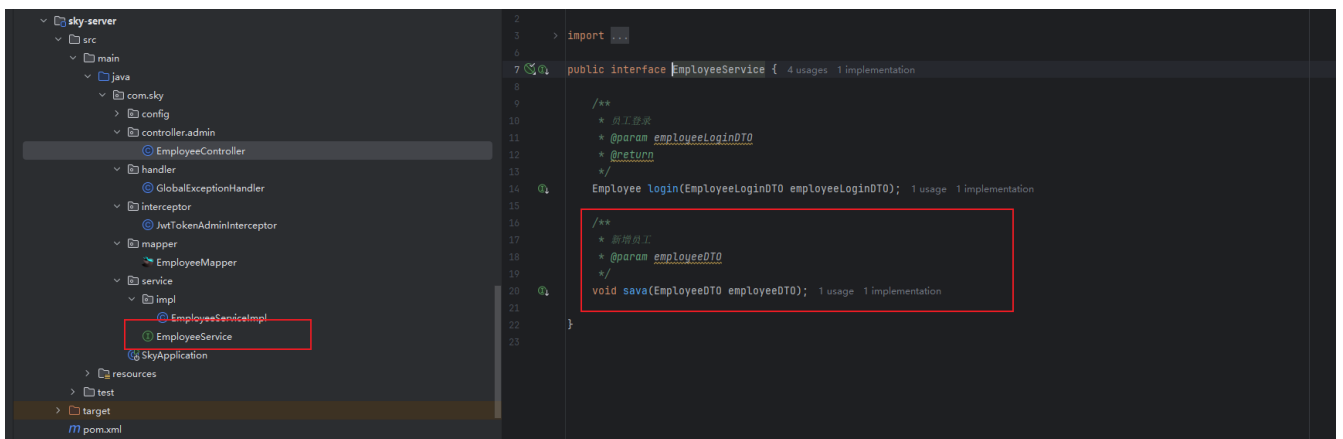
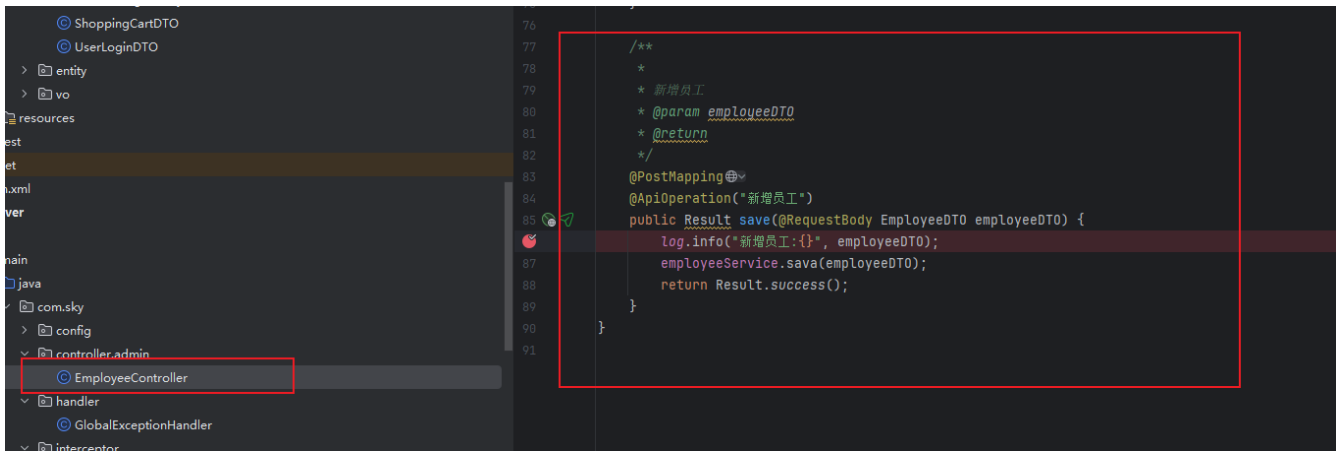


4.功能点一：完善登录功能，对密码进行MD5加密

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



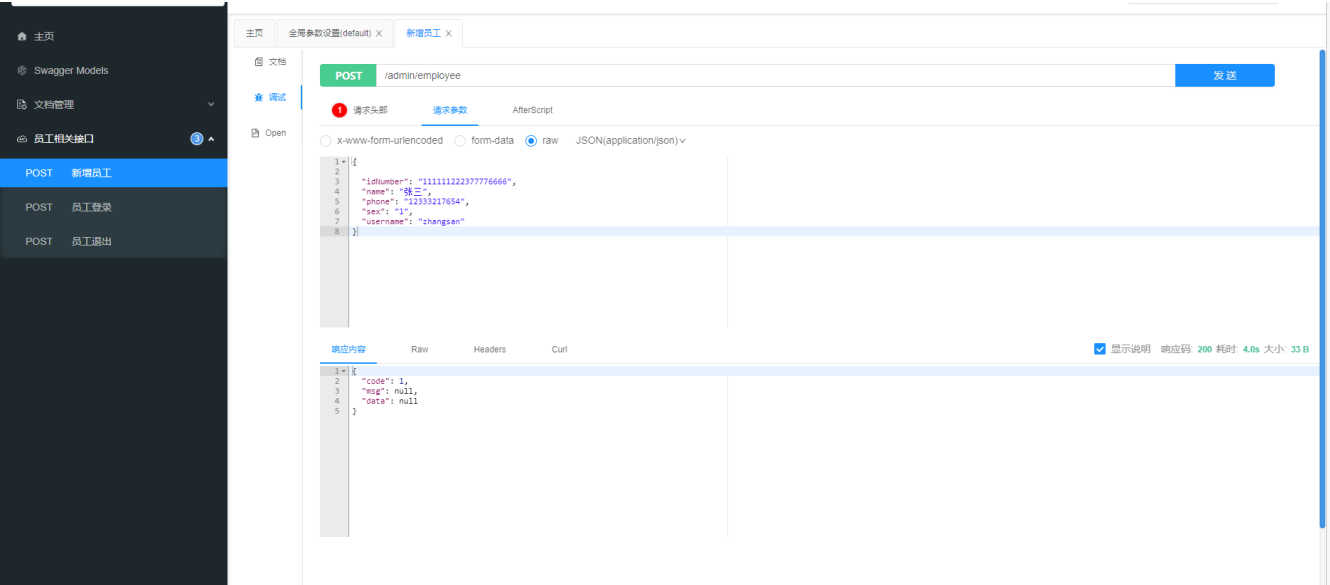
5.功能点二：新增员工功能实现



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

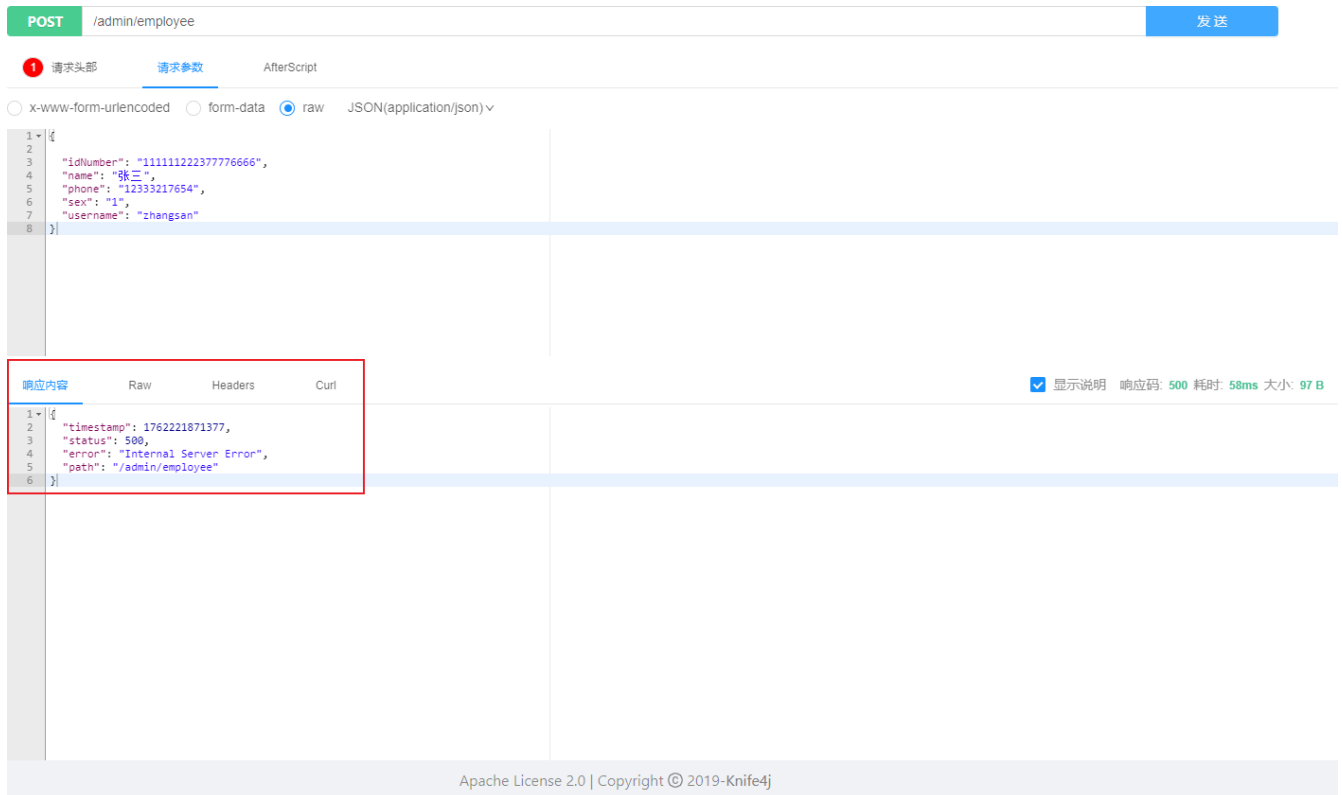


5.1如何使用异常处理器捕获异常

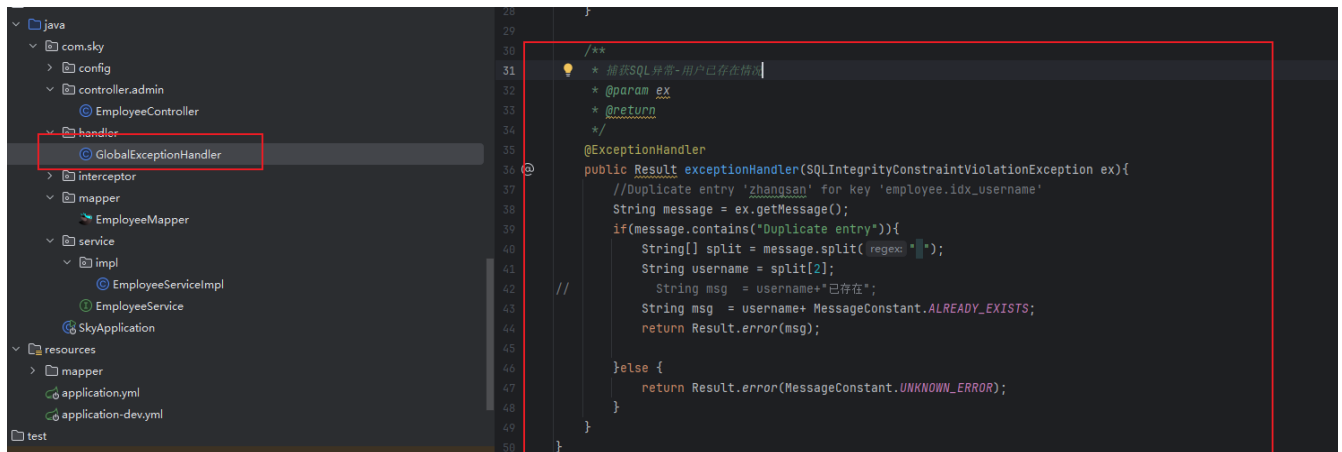
当插入重复用户时，报以下错误：

```
java.sql.SQLIntegrityConstraintViolationException Create breakpoint : Duplicate entry 'zhangsan' for key 'employee.idx_username'
at com.mysql.cj.jdbc.exceptions.SQLError.createSQLException(SQLError.java:117) ~[mysql-connector-java-8.0.30.jar:8.0.30]
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122) ~[mysql-connector-java-8.0.30.jar:8.0.30]
at com.mysql.cj.jdbc.ClientPreparedStatement.executeInternal(ClientPreparedStatement.java:916) ~[mysql-connector-java-8.0.30.jar:8.0.30]
at com.mysql.cj.jdbc.ClientPreparedStatement.execute(ClientPreparedStatement.java:354) ~[mysql-connector-java-8.0.30.jar:8.0.30]
> at com.alibaba.druid.pool.DruidPooledPreparedStatement.execute(DruidPooledPreparedStatement.java:497) ~[druid-1.2.1.jar:1.2.1] <3 internal lines>
><71 folded frames>
```

接口测试提示以下信息：



根据报错的异常类，配置该异常类的捕获方法，当出现该异常，并且判断如果错误信息里含有Duplicate entry，则为用户已存在的报错，写出下列msg提示信息返回。



再次接口测试：

1 请求头部

请求参数

AfterScript

☐ x-www-form-urlencoded

☐ form-data

☒ raw

JSON(application/json) v

1 {

2

3 "idNumber": "11111122237776666",

4 "name": "张三",

5 "phone": "12333217654",

6 "sex": "1",

7 "username": "zhangsan"

8 }

响应内容

Raw

Headers

Curl

1 {

2 "code": 0,

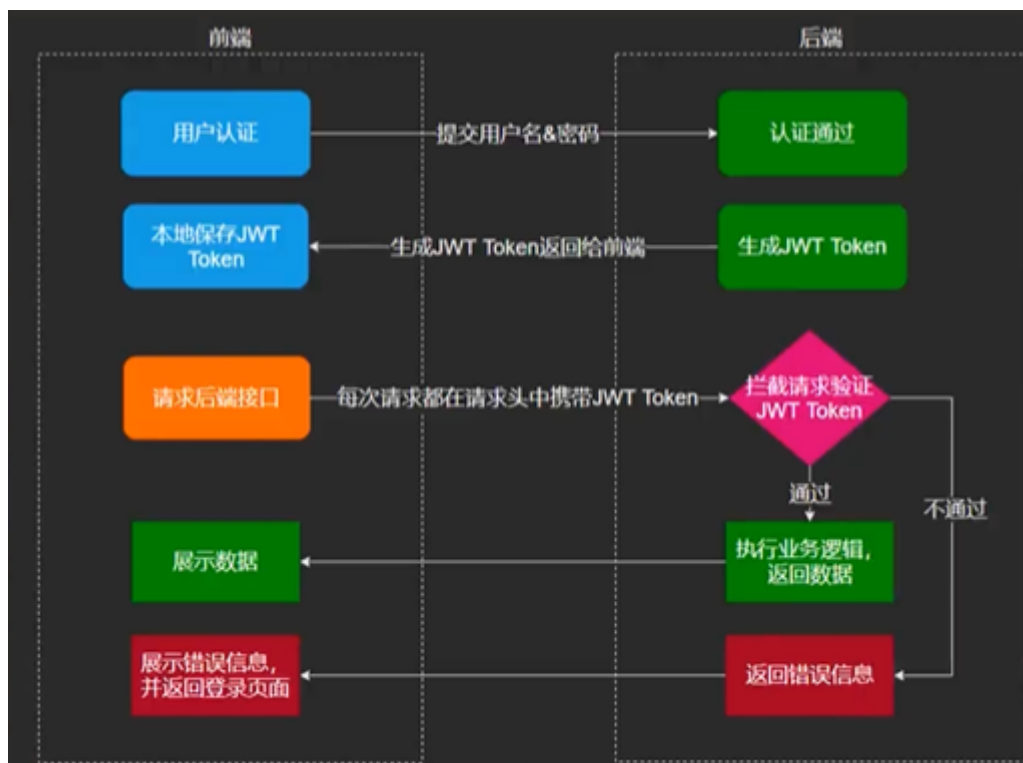
3 "msg": "'zhangsan'已存在",

4 "data": null

5 }

msg里已出现对应的提示信息。

5.2了解JWT流程



在新增的时候，想获取jwt里的用户id，

同一个请求，它的拦截，调用方法等等，都是在一个线程里，（可以通过在登录，jwt验证，新增员工代码上加入 `System.out.println("当前线程的id:"+Thread.currentThread().getId());`进行验证）

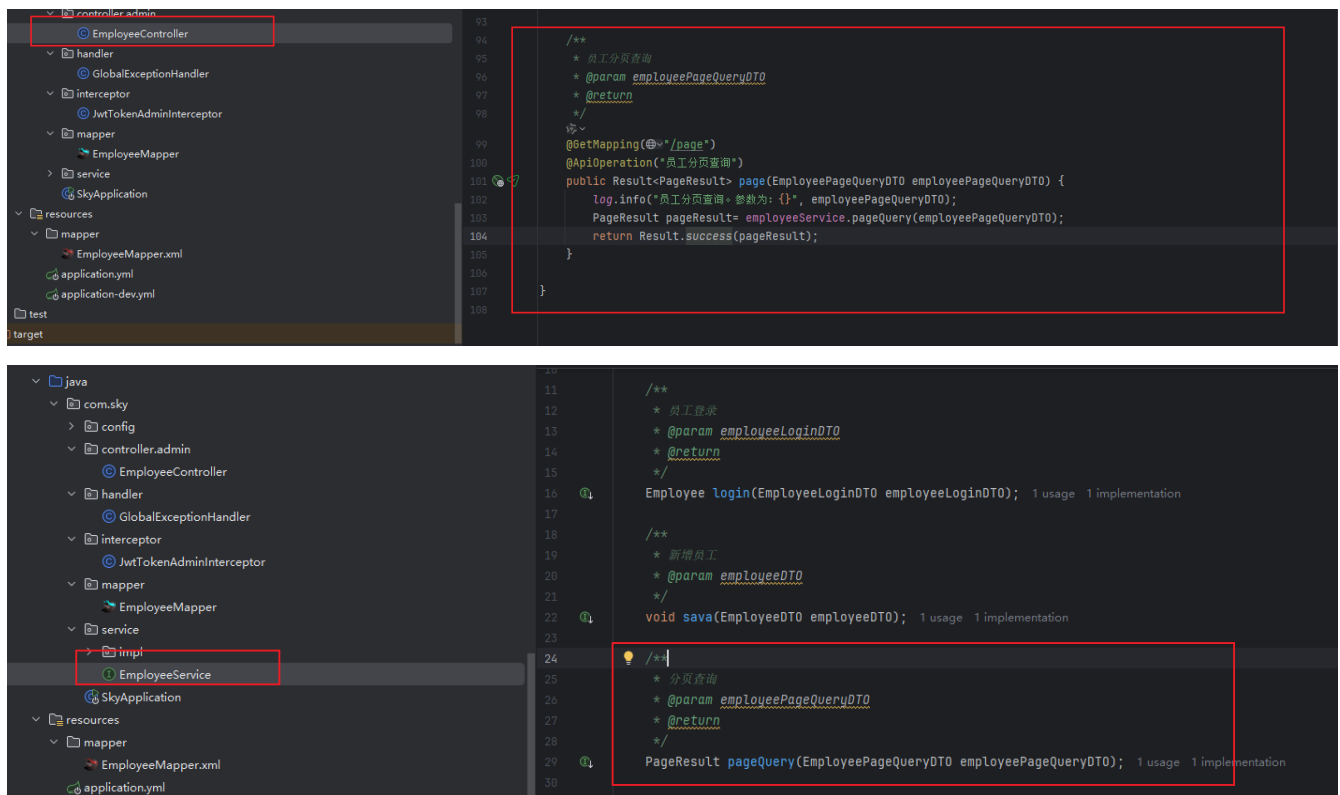
`ThreadLocal`就是`Thread`的局部变量，`ThreadLocal`为每个线程提供单独一份存储空间，具有线程隔离效果，只有在线程内才能获取到对应的值，线程外不能直接访问

`ThreadLocal`常用方法：

- `public void set(T value)` 设置当前线程的线程局部变量的值
- `public T get()` 返回当前线程所对应的线程局部变量的值
- `public void remove()` 移除当前线程的线程局部变量

6.功能点三：查询员工分页

6.1分页功能实现



```
public class EmployeeServiceImpl implements EmployeeService {  
    /**  
     * 分页查询  
     * @param employeePageQueryDTO  
     * @return  
     */  
    @Override  
    public PageResult pageQuery(EmployeePageQueryDTO employeePageQueryDTO) {  
        //select * from employee limit 0,10  
        //开始分页查询  
        PageHelper.startPage(employeePageQueryDTO.getPage(), employeePageQueryDTO.getPageSize());  
  
        Page<Employee> page = employeeMapper.pageQuery(employeePageQueryDTO);  
        long total = page.getTotal();  
        List<Employee> records = page.getResult();  
        PageResult pageResult = new PageResult(total, records);  
        return pageResult;  
    }  
}
```

```
<insert id="insertEmployee" parameterType="com.sky.entity.Employee">  
    <sql>  
        insert into employee (name, username, password, phone, sex, id_number, status, create_time)  
        values (#{name}, #{username}, #{password}, #{phone}, #{sex}, #{idNumber}, #{status}, #{createTime})  
    </sql>  
</insert>  
  
/**  
 * 分页查询  
 * @param employeePageQueryDTO  
 * @return  
 */  
Page<Employee> pageQuery(EmployeePageQueryDTO employeePageQueryDTO);
```

```
<select id="pageQuery" resultType="com.sky.entity.Employee">  
    select * from employee  
    <where>  
        <if test="name != null and name != ''">  
            and name like concat('%',#{name},'%')  
        </if>  
    </where>  
    order by create_time desc  
</select>  
</mapper>
```

6.2如何调整日期格式

方法一：在属性上加注解，对日期格式化

```
private String idNumber;  
  
private Integer status;  
  
@JsonFormat(pattern = "yyyy-MM-dd HH:mm:ss")  
private LocalDateTime createTime;  
  
@JsonFormat(pattern = "yyyy-MM-dd HH:mm:ss")  
private LocalDateTime updateTime;  
  
private Long createUser;  
  
private Long updateUser;  
}
```

☒	name	
☒	page	1
☒	pageSize	2

响应内容

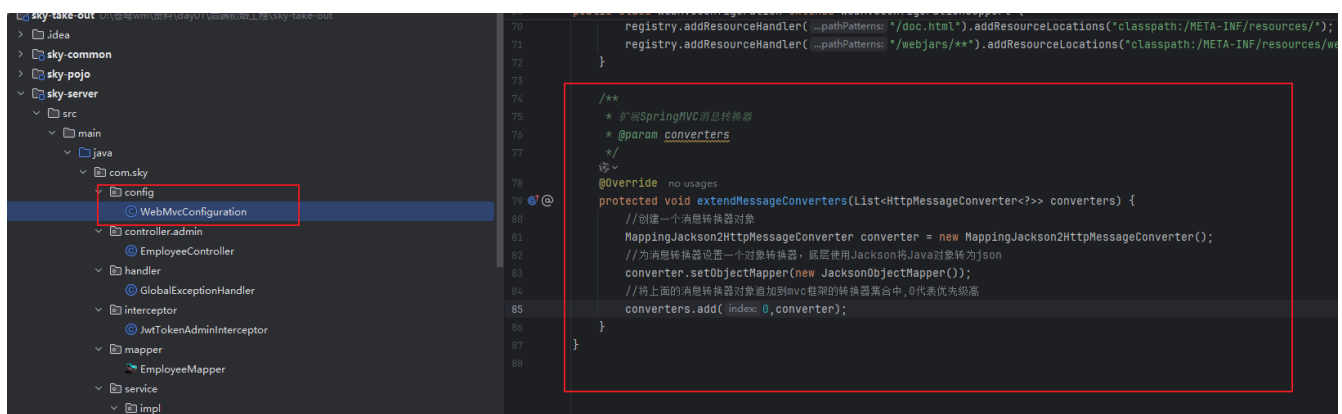
Raw

Headers

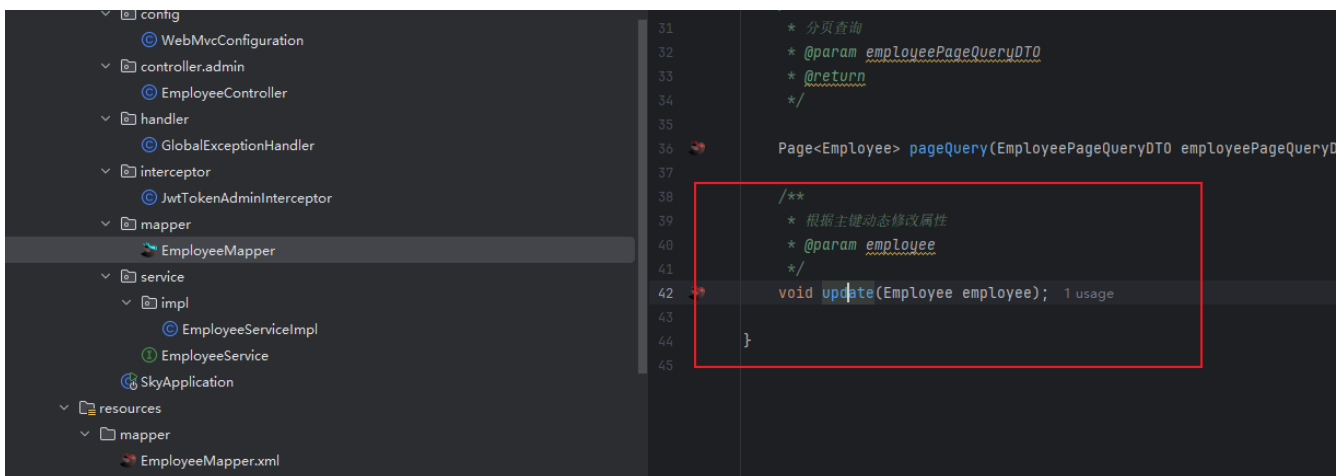
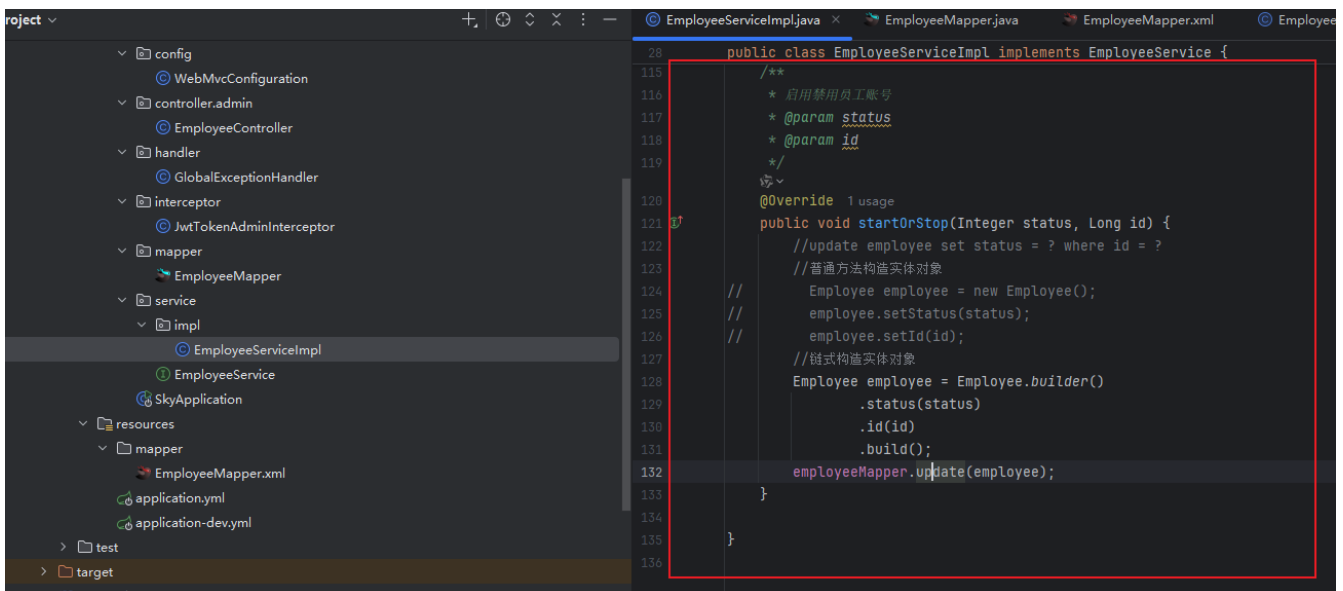
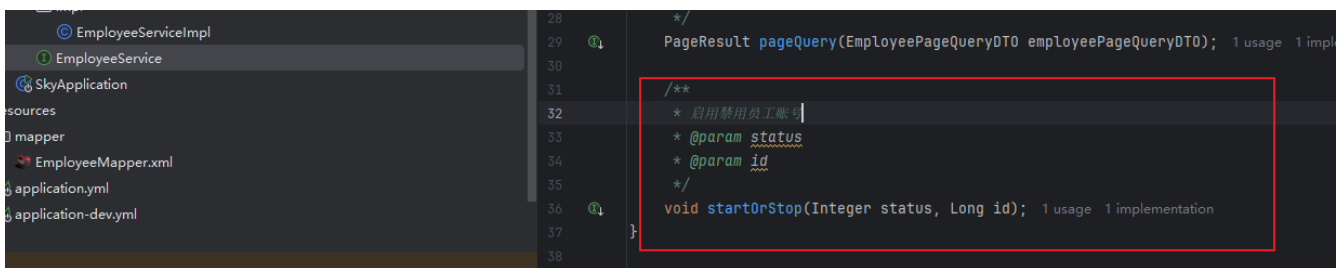
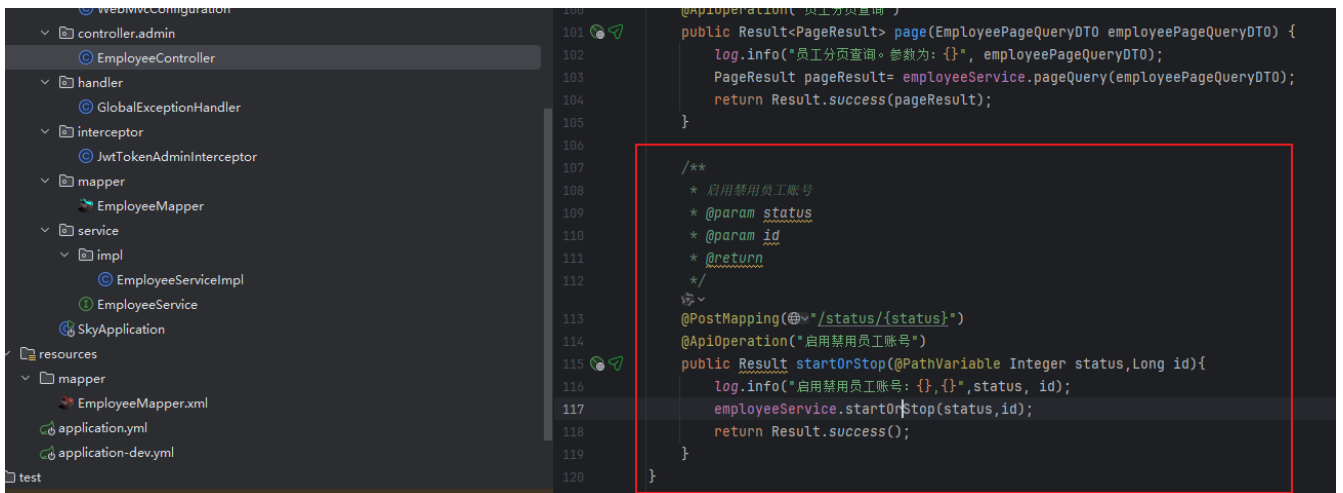
Curl

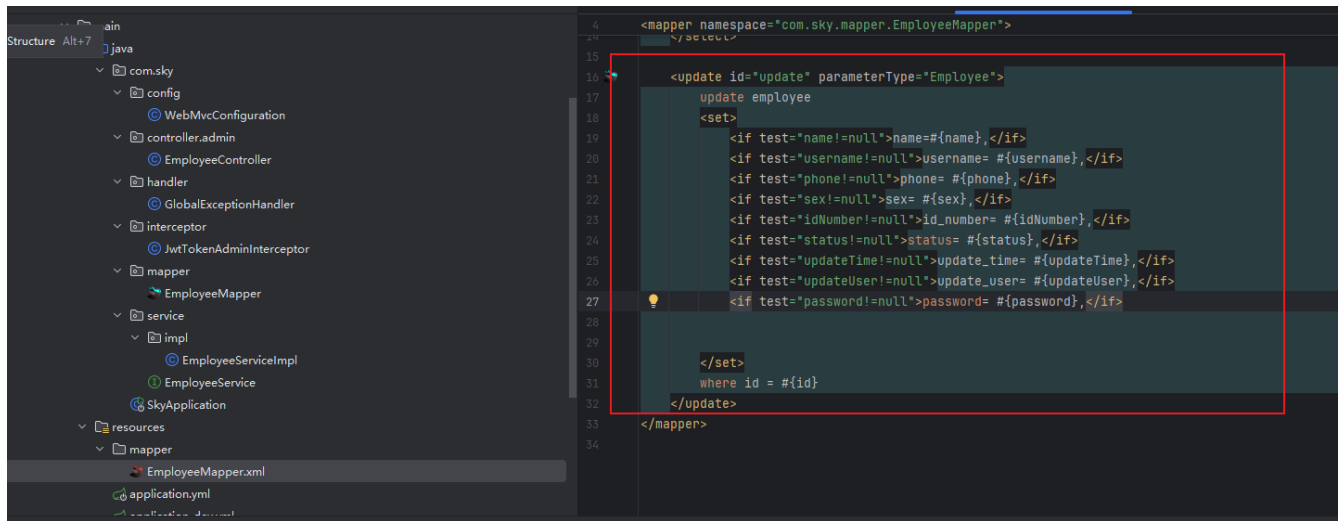
```
1 | {
2 |   "code": 1,
3 |   "msg": null,
4 |   "data": {
5 |     "total": 5,
6 |     "records": [
7 |       {
8 |         "id": 14,
9 |         "username": "zhaoliu",
10 |        "name": "赵六",
11 |        "password": "e10adc3949ba59abbe56e057f20f883e",
12 |        "phone": "13376547765",
13 |        "sex": "1",
14 |        "idNumber": "354785444125587745",
15 |        "status": 1,
16 |        "createTime": "2025-11-04 11:18:54",
17 |        "updateTime": "2025-11-04 11:18:54",
18 |        "createUser": 1,
19 |        "updateUser": 1
20 |      },
21 |      {
22 |        "id": 13,
23 |        "username": "wangwu",
24 |        "name": "王五",
25 |        "password": "e10adc3949ba59abbe56e057f20f883e",
26 |        "phone": "13744876434",
27 |        "sex": "1",
28 |        "idNumber": "371221222129332284",
29 |        "status": 1,
30 |        "createTime": "2025-11-04 11:16:47",
31 |        "updateTime": "2025-11-04 11:16:47",
32 |        "createUser": 1,
33 |        "updateUser": 1
34 |      }
35 |    ]
36 |  }
37 | }
```

方法二：在WebMvcConfiguration中扩展SpringMVC的消息转换器，统一对日期类型进行格式化处理



7.功能点四： 启用禁用员工





8.功能点五：编辑员工

