

### 3-9 实现注册超级管理员功能（业务层）

140.143.132.225:8000/project-1/doc-39

上一小节，我们封装了注册用户的持久层代码，下面就应该编写业务层的代码可。比如保存用户记录之前，我们要获得OpenId才行。

#### 一、获取OpenId

获取微信用户的 `OpenId`，需要后端程序向微信平台发出请求，并上传若干参数，最终才能得到。

#### URL请求路径

`https://api.weixin.qq.com/sns/jscode2session`

#### 请求参数

参数	必填	说明
appid	是	小程序唯一标识
secret	是	小程序密钥
js_code	是	临时授权字符串
grant_type	是	authorization_code

#### 返回参数

参数	说明
openid	用户唯一标识
session_key	会话密钥
session_key	用户在开放平台的唯一标识符

在 `com.example.emos.wx.service` 中创建 `UserService.java` 接口

```
1. public interface UserService {  
2. }
```

在 `com.example.emos.wx.service.impl` 中创建 `UserServiceImpl.java` 类

```
1. package com.example.emos.wx.service.impl;
2. .....
3. @Service
4. @Slf4j
5. @Scope("prototype")
6. public class UserServiceImpl implements UserService {
7.     @Value("${wx.app-id}")
8.     private String appId;
9.     @Value("${wx.app-secret}")
10.    private String appSecret;
11.    @Autowired
12.    private TbUserDao userDao;
13.    private String getOpenId(String code) {
14.        String url = "https://api.weixin.qq.com/sns/jscode2session";
15.        HashMap map = new HashMap();
16.        map.put("appid", appId);
17.        map.put("secret", appSecret);
18.        map.put("js_code", code);
19.        map.put("grant_type", "authorization_code");
20.        String response = HttpUtil.post(url, map);
21.        JSONObject json = JSONUtil.parseObj(response);
22.        String openId = json.getStr("openid");
23.        if (openId == null || openId.length() == 0) {
24.            throw new RuntimeException("临时登陆凭证错误");
25.        }
26.        return openId;
27.    }
28. }
```

## 二、编写注册新用户的业务代码

在 `UserService` 接口中添加抽象方法的声明

```
1. public int registerUser(String registerCode,String code,String nickname,String photo);
```

在 `ServiceImpl` 类中实现抽象方法

```
1. @Override
2. public int registerUser(String registerCode, String code, String nickname, String photo) {
3.     //如果邀请码是000000, 代表是超级管理员
4.     if (registerCode.equals("000000")) {
5.         //查询超级管理员帐户是否已经绑定
6.         boolean bool = userDao.haveRootUser();
7.         if (!bool) {
8.             //把当前用户绑定到ROOT帐户
9.             String openId = getOpenId(code);
10.            HashMap param = new HashMap();
11.            param.put("openId", openId);
12.            param.put("nickname", nickname);
13.            param.put("photo", photo);
14.            param.put("role", "[0]");
15.            param.put("status", 1);
16.            param.put("createTime", new Date());
17.            param.put("root", true);
18.            userDao.insert(param);
19.            int id = userDao.searchIdByOpenId(openId);
20.            return id;
21.        } else {
22.            //如果root已经绑定了, 就抛出异常
23.            throw new EmosException("无法绑定超级管理员账号");
24.        }
25.    }
26.    //TODO 此处还有其他判断内容
27.    else{
28.        return 0;
29.    }
30. }
```