3-10 实现注册超级管理员功能（Web层）

140.143.132.225:8000/project-1/doc-41

一、创建表单类

接收移动端提交的注册请求，我们需要用表单类来封装数据，所以创建 `RegisterForm.java` 类。

```
package com.example.emos.wx.controller.form;
import io.swagger.annotations.ApiModel;
import lombok.Data;
import javax.validation.constraints.NotBlank;
import javax.validation.constraints.Pattern;
@Data
@ApiModel
public class RegisterForm {
    @NotBlank(message = "注册码不能为空")
    @Pattern(regexp = "^[0-9]{6}$",message = "注册码必须是6位数字")
    private String registerCode;
    @NotBlank(message = "微信临时授权不能为空")
    private String code;
    @NotBlank(message = "昵称不能为空")
    private String nickname;
    @NotBlank(message = "头像不能为空")
    private String photo;
}
```

二、创建Controller类

处理移动端提交的请求，我们需要Controller类，所以创建 `UserController.java` 类。

> 问：业务层采用先定义接口，后声明实现类的做法，为什么Web层不这么做？
> 答：业务层的需求经常变化，所以应该先声明接口，然后再写实现类。Web层这里变化并不大，可以直接定义具体类。

```
package com.example.emos.wx.controller;
……
@RestController
@RequestMapping("/user")
@Api("用户模块Web接口")
public class UserController {
    @Autowired
    private UserService userService;
    @Autowired
    private JwtUtil jwtUtil;
    @Autowired
    private RedisTemplate redisTemplate;
    @Value("${emos.jwt.cache-expire}")
    private int cacheExpire;
    @PostMapping("/register")
    @ApiOperation("注册用户")
    public R register(@Valid @RequestBody RegisterForm form) {
        int id = userService.registerUser(form.getRegisterCode(), form.getCode(),
form.getNickname(), form.getPhoto());
        String token = jwtUtil.createToken(id);
        Set<String> permsSet = userService.searchUserPermissions(id);
        saveCacheToken(token, id);
        return R.ok("用户注册成功").put("token", token).put("permission", permsSet);
    }
    private void saveCacheToken(String token, int userId) {
        redisTemplate.opsForValue().set(token, userId + "", cacheExpire, TimeUnit.DAYS);
    }
}
```