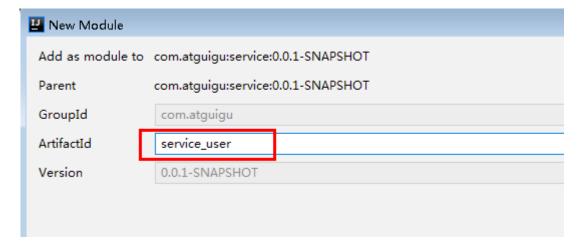
一、登录需求分析



- 1, 登录采取弹出层的形式
- 2, 登录方式:
 - (1) 手机号码+手机验证码
 - (2) 微信扫描
- 3,无注册界面,第一次登录根据手机号判断系统是否存在,如果不存在则自动注册
- 4, 微信扫描登录成功必须绑定手机号码,即:第一次扫描成功后绑定手机号,以后登录扫描直接登录成功
- 5, 网关统一判断登录状态, 如何需要登录, 页面弹出登录层

二、搭建service-user模块

1、在service下创建service_user模块



2、添加application.properties

```
1 # 服务端口
 2 server.port=8203
 3 # 服务名
4 spring.application.name=service-user
 6 # 环境设置: dev、test、prod
 7 spring.profiles.active=dev
9 # mysql数据库连接
10 spring.datasource.driver-class-name=com.mysql.jdbc.Driver
11 spring.datasource.url=jdbc:mysql://localhost:3306/yygh_user?characterEncoding
12 spring.datasource.username=root
13 spring.datasource.password=root
14
15 #返回json的全局时间格式
16 spring.jackson.date-format=yyyy-MM-dd HH:mm:ss
17 spring.jackson.time-zone=GMT+8
18
19 # nacos服务地址
20 spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848
21
22 #配置mapper xml文件的路径
23 mybatis-plus.mapper-locations=classpath:com/atguigu/yygh/user/mapper/xml/*.xm
```

3、创建数据库和表

4、创建启动类

```
1  @SpringBootApplication
2  @ComponentScan(basePackages = "com.atguigu")
3  @EnableDiscoveryClient
4  @EnableFeignClients(basePackages = "com.atguigu")
5  @MapperScan("com.atguigu.yygh.user.mapper")
6  public class ServiceUserApplication {
7    public static void main(String[] args) {
8         SpringApplication.run(ServiceUserApplication.class, args);
9    }
10 }
```

5、配置GateWay网关

```
1 #设置路由id
2 spring.cloud.gateway.routes[3].id=service-user
3 #设置路由的uri
4 spring.cloud.gateway.routes[3].uri=lb://service-user
5 #设置路由断言,代理servicerId为auth-service的/auth/路径
6 spring.cloud.gateway.routes[3].predicates= Path=/*/user/**
```

三、登录接口实现

1、添加service接口与实现

1.1 创建UserInfoService定义方法

```
1 public interface UserInfoService extends IService<UserInfo> {
2    //会员登录
3    Map<String, Object> login(LoginVo loginVo);
4 }
```

1.2 创建UserInfoServiceImpl实现方法

```
1 @Service
  public class UserInfoServiceImpl extends
 3
           ServiceImpl<UserInfoMapper, UserInfo> implements UserInfoService {
      @Override
 4
       public Map<String, Object> login(LoginVo loginVo) {
 5
           String phone = loginVo.getPhone();
 6
 7
           String code = loginVo.getCode();
           //校验参数
 8
           if(StringUtils.isEmpty(phone) ||
9
                   StringUtils.isEmpty(code)) {
10
               throw new YyghException(20001,"数据为空");
11
           }
13
14
           //TODO 校验校验验证码
15
           //手机号已被使用
16
           QueryWrapper<UserInfo> queryWrapper = new QueryWrapper<>();
17
           queryWrapper.eq("phone", phone);
18
19
           //获取会员
           UserInfo userInfo = baseMapper.selectOne(queryWrapper);
20
           if(null == userInfo) {
21
22
               userInfo = new UserInfo();
23
               userInfo.setName("");
               userInfo.setPhone(phone);
24
               userInfo.setStatus(1);
25
               this.save(userInfo);
26
27
           }
           //校验是否被禁用
28
           if(userInfo.getStatus() == 0) {
29
30
               throw new YyghException(20001,"用户已经禁用");
31
           }
32
           //返回页面显示名称
33
34
           Map<String, Object> map = new HashMap<>();
           String name = userInfo.getName();
35
           if(StringUtils.isEmpty(name)) {
36
               name = userInfo.getNickName();
37
38
           }
```

```
if(StringUtils.isEmpty(name)) {
    name = userInfo.getPhone();
}

map.put("name", name);

map.put("token", "");

return map;
}
```

2、添加Mapper接口

```
public interface UserInfoMapper extends BaseMapper<UserInfo> {
}
```

3、添加Controller方法

3.1 创建UserInfoApiController添加方法

```
1 @RestController
 2 @RequestMapping("/api/user")
 3 public class UserInfoApiController {
 4
       @Autowired
       private UserInfoService userInfoService;
 6
       @ApiOperation(value = "会员登录")
 8
9
       @PostMapping("login")
       public R login(@RequestBody LoginVo loginVo) {
10
11
           Map<String, Object> info = userInfoService.login(loginVo);
12
           return R.ok().data(info);
       }
13
14 }
```