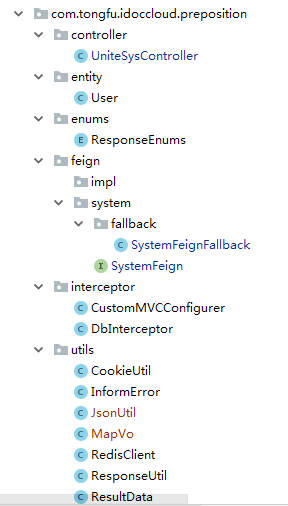
**iDoc Cloud 后端开发规范**

# 后端开发工具

统一： IntelliJ IDEA

优点：代码提示功能强大、开发效率高、插件多，可根据需要去下载插件（如热部署，代码规约）

# 项目结构



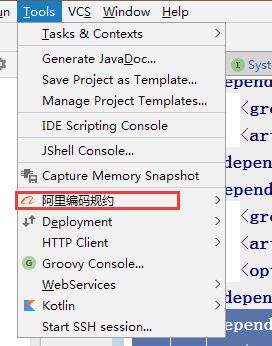
关键点：

1. 文件目录英文全是小写
2. 类名首字母大写，每个单词首字母大写
3. controller层书写接口，entity存放实体类，service写业务等等
4. 主要逻辑写在service实现类，controller主要实现调用方法

# 代码规范

代码规范以阿里编码规约为主，请务必规范书写。

IDEA引入插件。百度搜索idea 阿里编码规约有安装教程。



# 返回的固定json格式

作用：返回格式统一，方便前端获取数据。

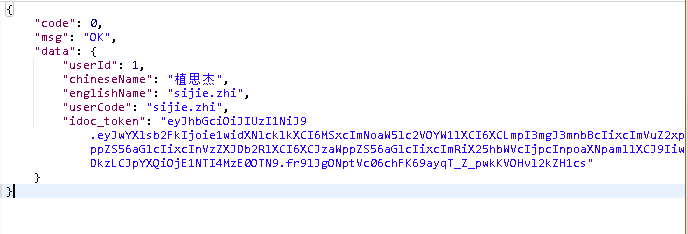
工具类已封装，请参考以下几个工具类：

ResponseUtil

ResponseVO

ResponseEnums

返回格式案例：



# 返回状态码定义

参考控制器响应枚举类：

ResponseEnums

*/\*\*  
 \* 0 成功  
 \*/****OK***(0, **"OK"**),  
*/\*\*  
 \* 400 参数错误  
 \*/****PARAM\_ERROR***(400, **"PARAM ERROR"**),  
*/\*\*  
 \*500 错误  
 \*/****ERROR***(500, **"ERROR"**),  
***NO\_DATA***(700, **"数据找不到了"**),  
***INSERT\_ERROR***(800,**"INSERT ERROR"**),  
***UPDATE\_ERROR***(801,**"UPDATE ERROR"**),  
***DELETE\_ERROR***(802,**"DELETE ERROR"**),

# 请求接口流程及参数规范

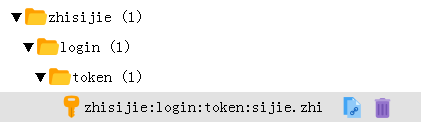
## 登录接口

用户登录成功后，Redis存放用户的格式是：

db\_name:login:token:userCode

如：

zhisijie:login:token:sijie.zhi



作用：区分客户公司，区分redis类型

## 其他接口（统一入口）

接口调用流程：

网关→前置服务→其他服务

除登录接口外，其他接口都统一调用前置服务一个接口，根据operation去调用其他服务。

网关处做了鉴权：

1）检测请求接口是否带上idoc\_token和operation 、ip、（物理网卡地址）

2）检测idoc\_token是否正确，及刷新时间

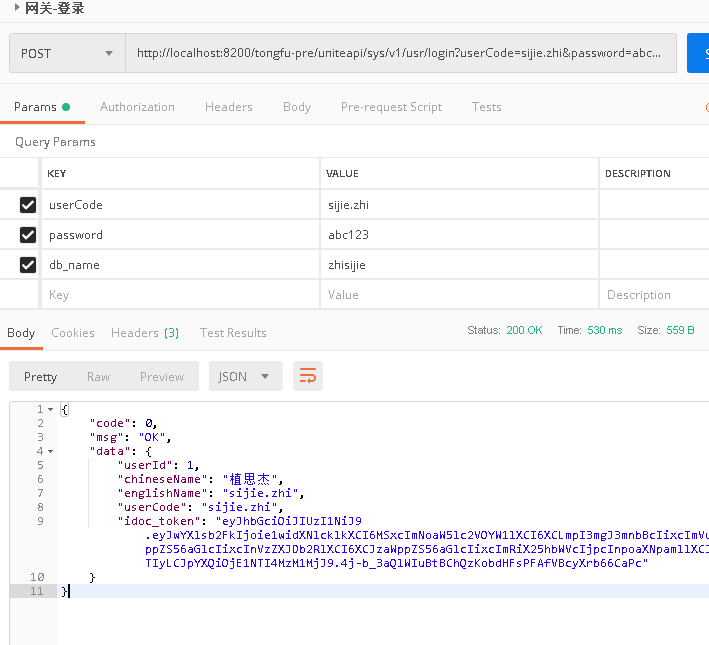
注意：必须带上idoc\_token 和 operation。token是在header，operation 是参数形式。

# 接口书写模板

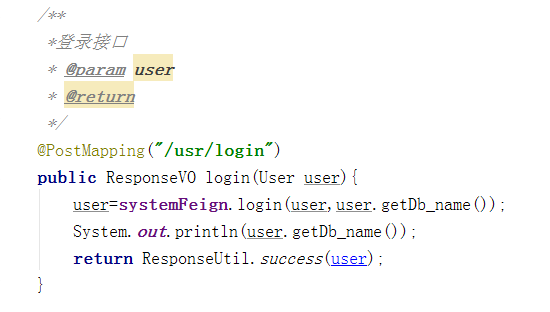
## 登录接口（非统一接口）

原因：登录接口不需要过多的鉴权，所以设单个接口

传入参数：用户名 密码 客户公司的数据库名

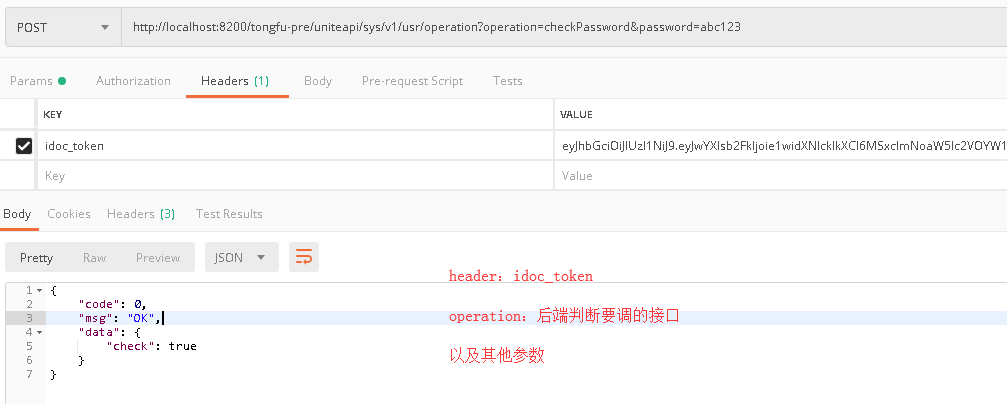


参数接受及调用接口：



## 其他接口（统一接口）

注意：idoc\_token 和 operation 不能少，否则鉴权不通过



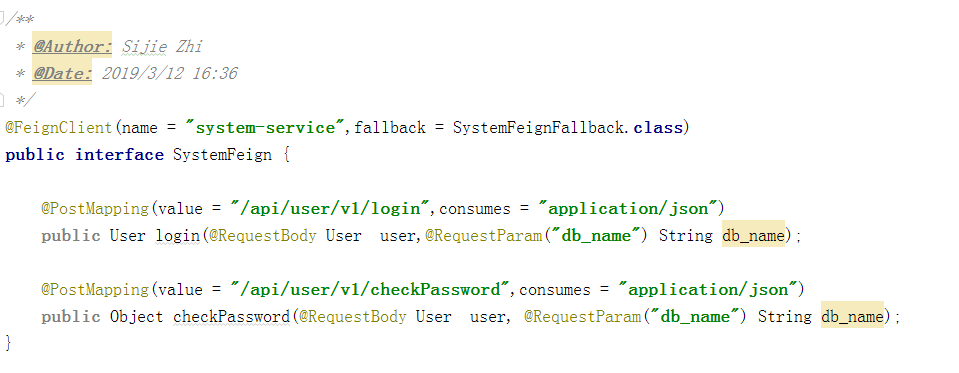
接口书写规范：



调用接口书写：

1. 参数以及方法名和被调用服务方法名一致
2. 要带上注解如@requestParam、@requestBody

注意:此处规范很重要，不懂的参数传递有不同的注解，前后一致。



调用失败熔断降级：

作用：防止微服务雪崩。所以每调用后置接口，都要写入一个降级接口。以通知维护人员。

