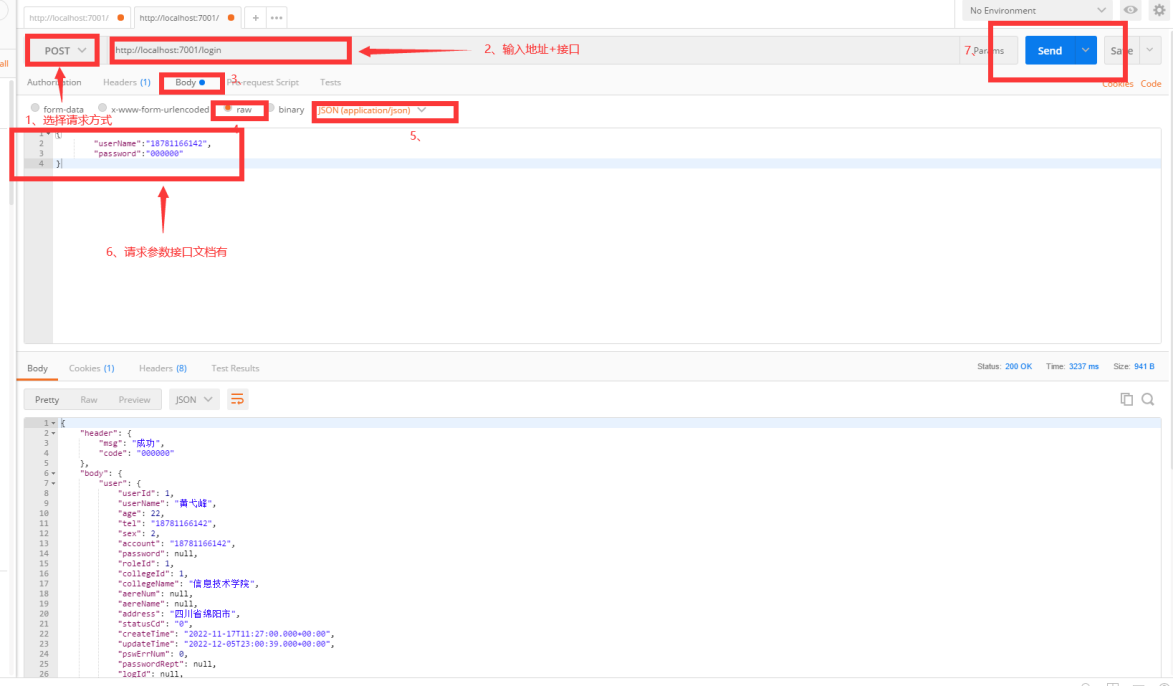
# 开发手册

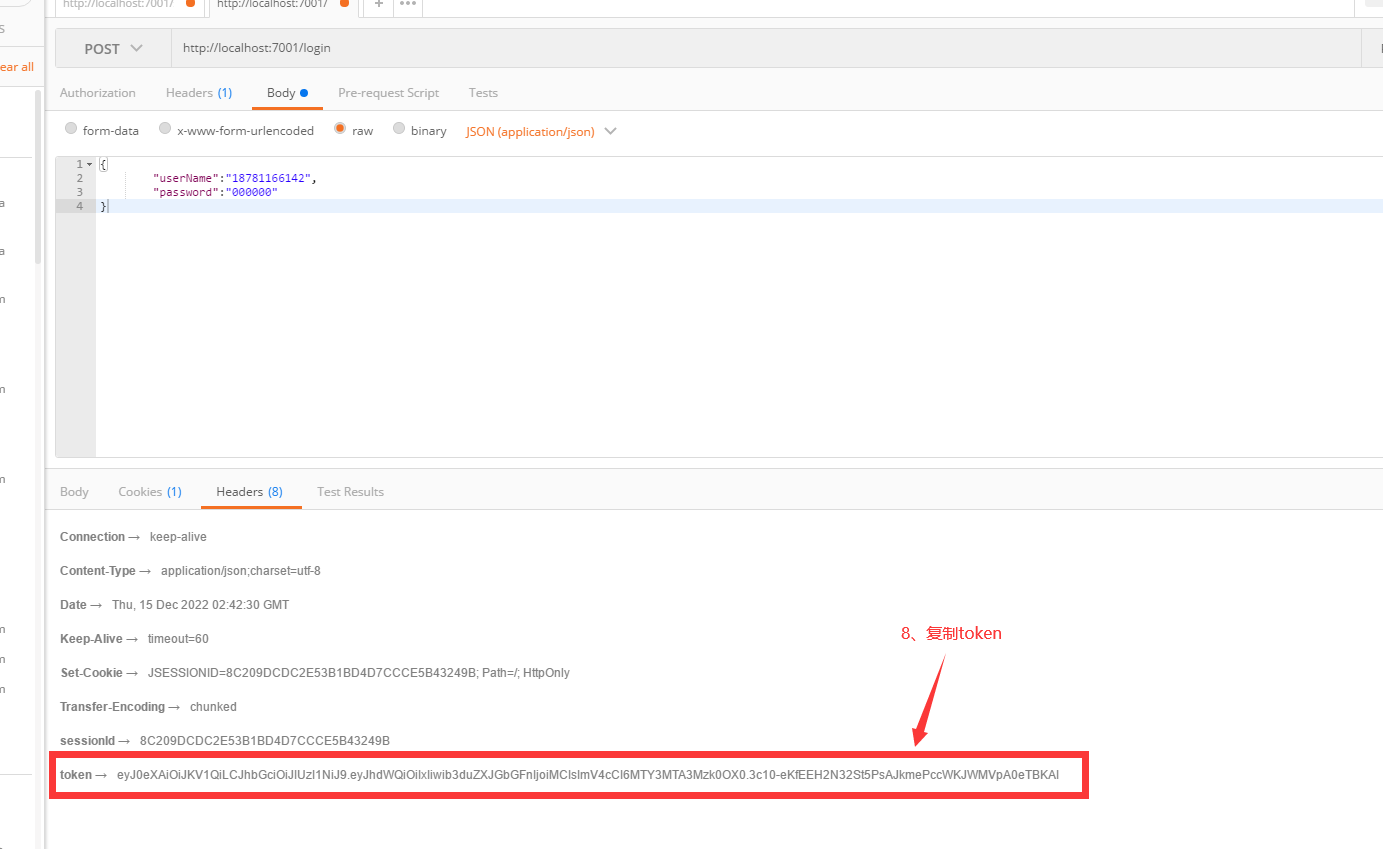
## 现有的登录账号密码

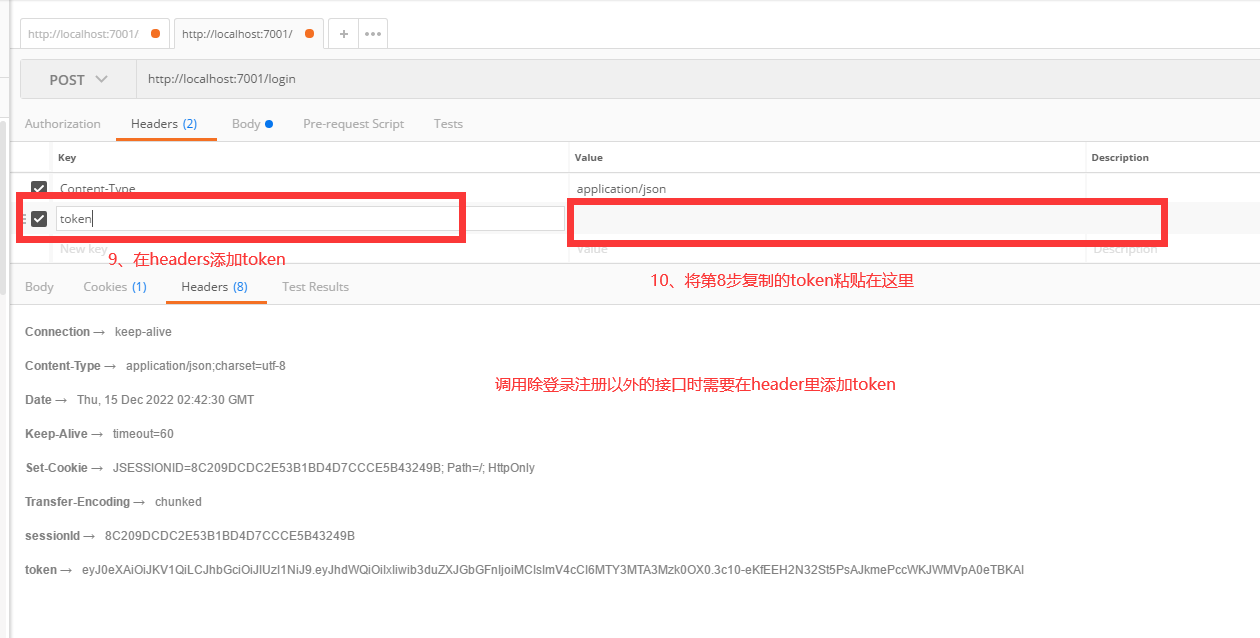
黄弋峰  账号：18781166142 密码：000000  
师延正  账号：18780000000 密码：000000

|  |  |
| --- | --- |
| 服务器地址 | [116.63.133.52:8080](http://116.63.133.52:8080" \o "http://116.63.133.52:8080) |
| 接口文档地址 | <http://116.63.133.52:8080/swagger-ui.html> |

## Postman使用方法





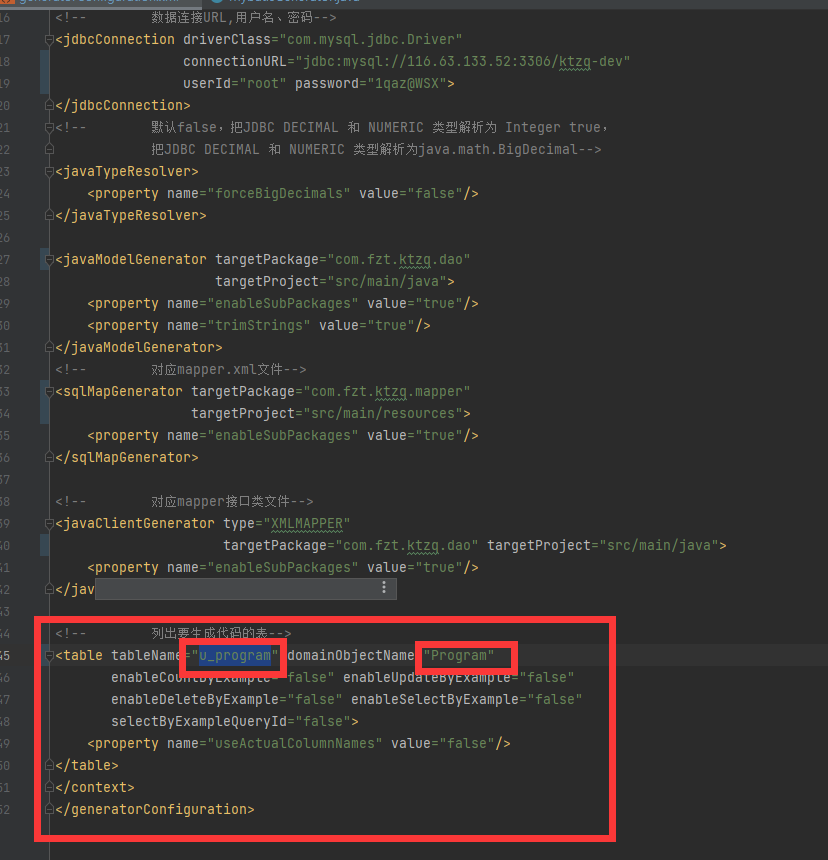


## 后端开发流程

摒弃了xml配置开发，为降低代码量采用的通用Mapper方式开发

### 3.1、代码生成器

在src/main/resources/generator/generatorConfiguration.xml下



启动



之后会在对应文件夹下生成表对应的实体类，接口

### 3.2、目录结构

ktzq-comm 公共方法类、常量类、工具类

ktzq-dao 实体类

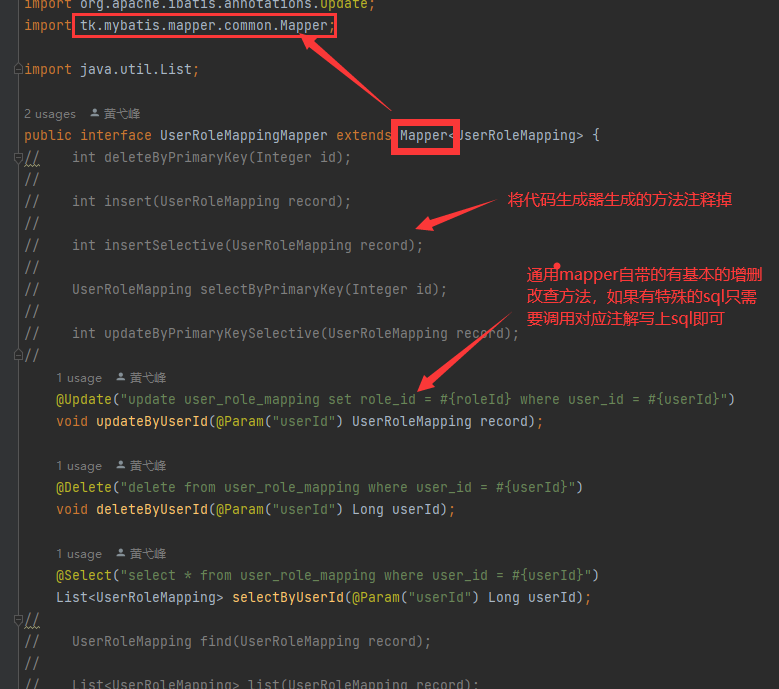
ktzq-mapper 接口类

ktzq-service 接口方法、控制类、系统配置

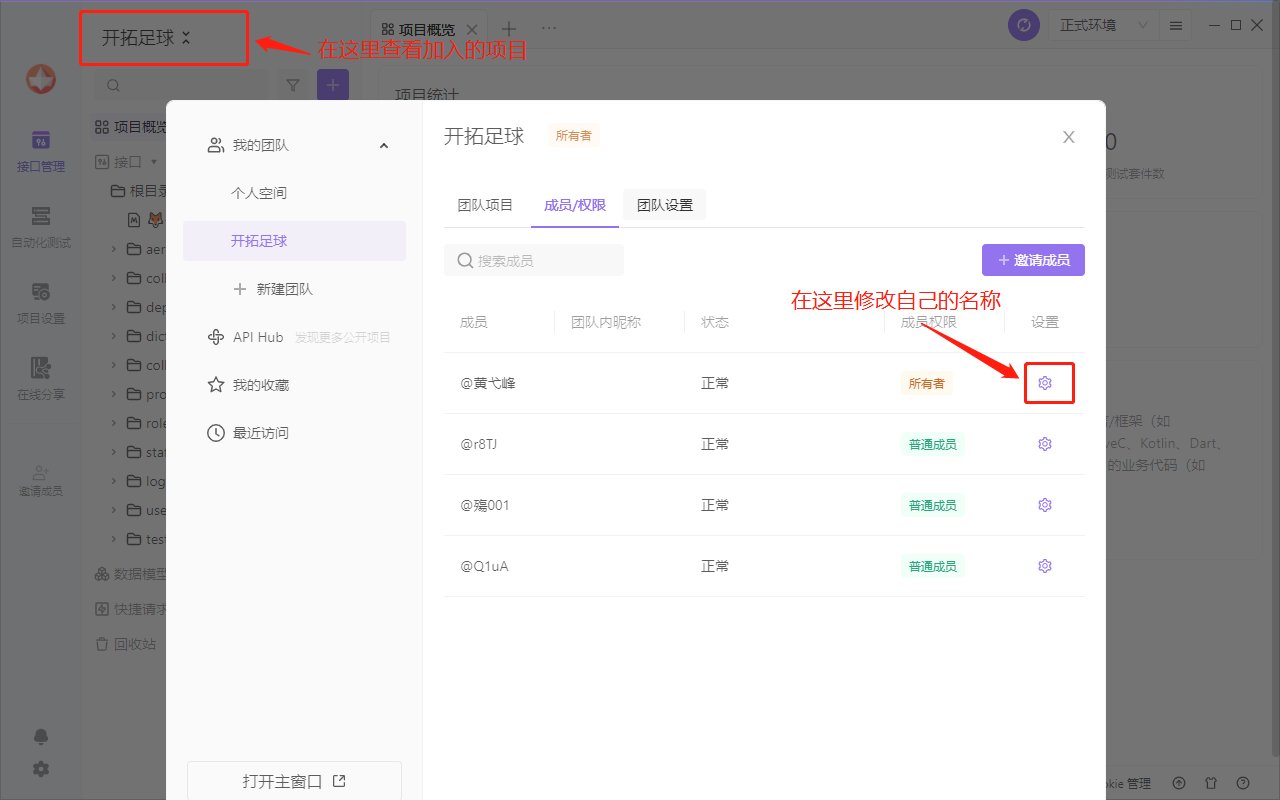
#### ktzq-dao

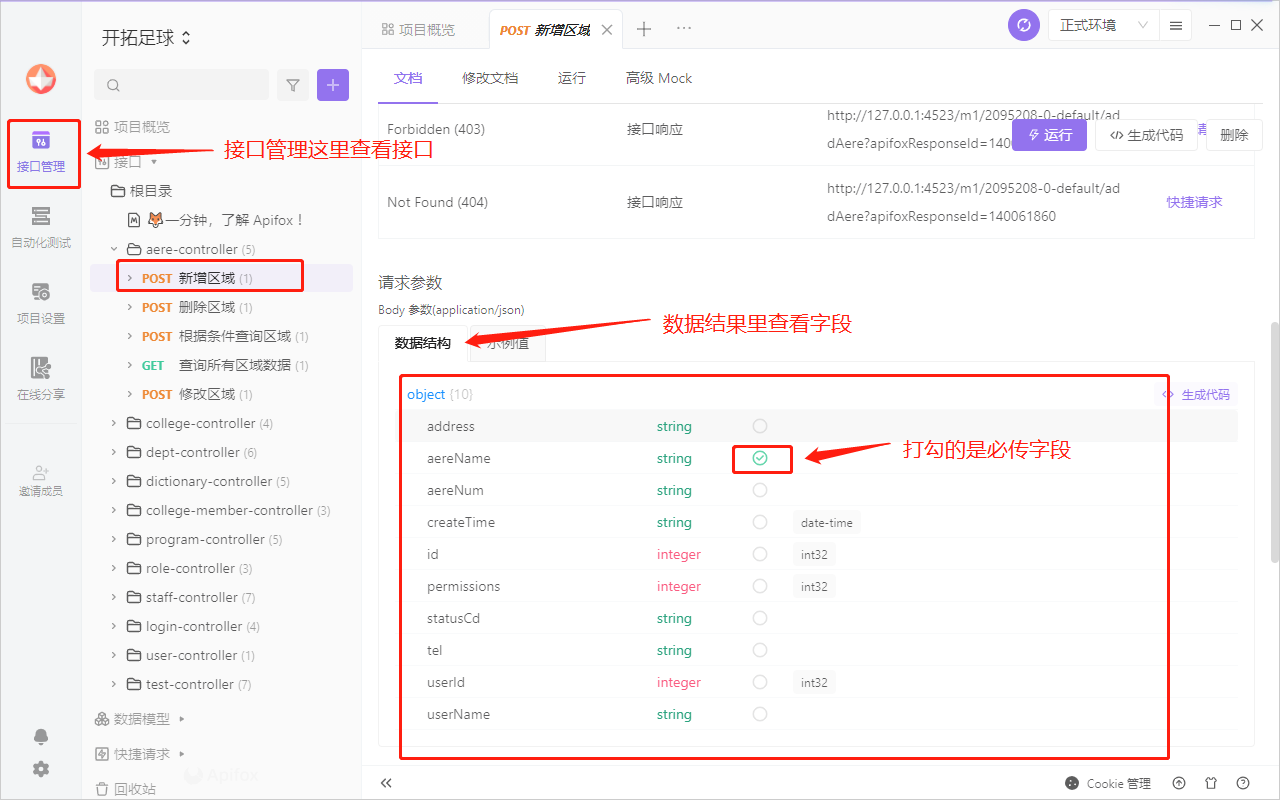


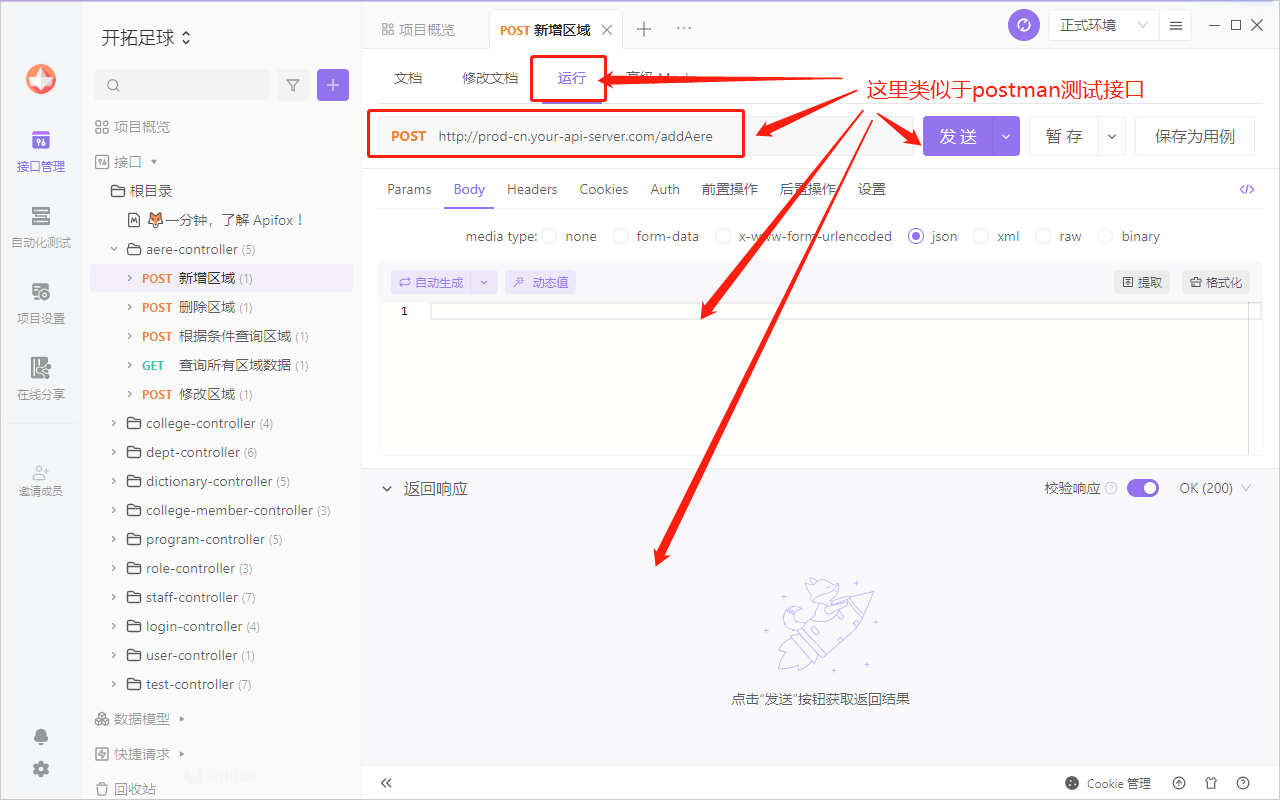
#### ktzq-mapper



## Apifox使用方法







## 数据脱敏

在接口返回数据前，在序列化的时候对敏感字段值进行处理，并且选用 jackson 的序列化来实现。

其中，自定义序列化器PrivacySerializer 是具体的实现过程，因为要脱敏的数据都是 String 类型的，所以继承 JsonSerializer 时的类型填 String。

重写的 serialize 方法是实现脱敏的核心，根据类型 type 的不同去设置序列化后的值。

重写的 createContextual 方法就是去读取我们自定义的 PrivacyEncrypt 注解，打造一个上下文的环境。

在自定义注解PrivacyEncrypt里，只有 type 的值为 PrivacyTypeEnum.CUSTOMER（自定义）时，才需要指定脱敏范围，即 prefixNoMaskLen 和 suffixNoMaskLen 的值，像邮箱、手机号这种隐藏格式都采用固定的

### 5.1、注解使用

直接在需要脱敏的字段上加上注解，指定 type 值即可，如下：

