**api接口开发流程**

# 搭建项目

## 新建项目

两个mvn项目，一个api项目，一个service项目

## service pom中引入

<parent>

<groupId>com.ubtechinc</groupId>

<artifactId>ubtechinc-parent</artifactId>

<version>1.0.0</version>

</parent>

<dependency>

<groupId>com.ubtechinc</groupId>

<artifactId>ubtechinc-core</artifactId>

<version>1.0.0</version>

</dependency>

<dependency>

<groupId>com.ubtechinc</groupId>

<artifactId>ubtechinc-common-utils</artifactId>

<version>1.0.0</version>

</dependency>

<dependency>

<groupId>com.ubtechinc</groupId>

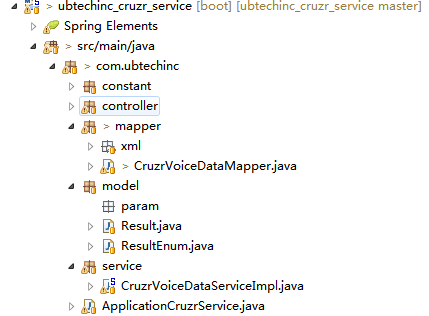
<artifactId>ubtechinc-authorization</artifactId>

<version>1.0.0</version>

</dependency>

## 3、创建项目包结构，如图所示

**service项目结构**



constant:存放常量类

controller：存放controller类

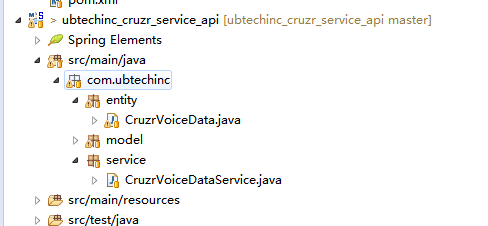
mapper：存放表对应的mapper接口类

mapper.xml:存放对应的xml文件

model：存放参数模型类和controller返回值模型类

service: 存放api service接口实现类

**api项目结构**



entity:表对应的实体类

sercie：提供服务的接口

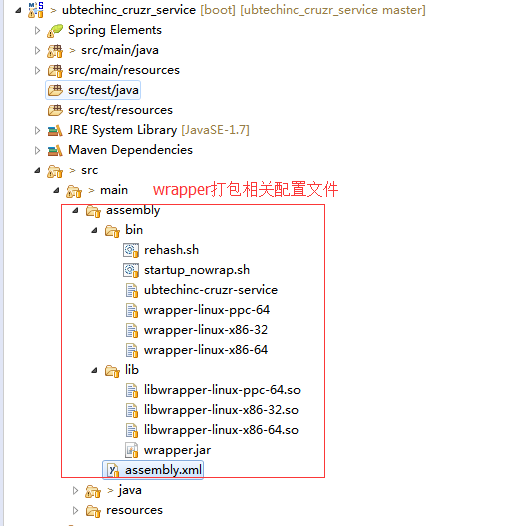
model：存放service返回值模型类

## 4、引入配置文件

**src/main/resource引入对应的properties文件**

1. application.properties spring系统配置文件
2. dubbo.properties dubbo配置文件
3. jedis.properties redis配置文件
4. druid.properties 数据库配置文件
5. wrapper.conf wrapper打包配置文件
6. 修改wrapper.conf配置文件的wrapper.app.parameter.1参数为对应的 aplication启动类

**src/main中引入对应的assembly配置信息**



## 5、配置信息

**1)在application启动类中配置需要扫描的类**

@MapperScan("com.ubtechinc.mapper\*")

1. **application.properties配置文件信息**

#服务启动端口，pom文件中获取

server.port=${server.port}

#mybatis扫描的配置文件

mybatis.mapper-locations=classpath:com/ubtechinc/mapper/xml/\*.xml

mybatis.typeAliasesPackage= com.ubtechinc.entity

#设置使用cglib动态代理

spring.aop.proxy-target-class=true

1. **druid.properties配置文件信息**

#数据库url，pom中获取

druid.url=${druid.url}

#数据库用户名，pom中获取

druid.username=${druid.username}

#数据库密码，pom中获取

druid.password=${druid.password}

druid.driver-class-name=com.mysql.jdbc.Driver

1. **dubbo.properties配置文件信息**

dubbo.application-name=redis-dubbo

dubbo.scan-package= com.ubtechinc.service

dubbo.protocol-name=dubbo

dubbo.protocol-accessLog=true

#协议端口，pom中获取

dubbo.protocol-port=${dubbo.registry.port}

dubbo.provider-timeout=3000

dubbo.provider-retries=1

dubbo.provider-delay=-1

dubbo.registry-protocol=zookeeper

#注册机地址，pom中获取

dubbo.registry-address=${dubbo.registry.address}

1. **jedis.properties配置文件信息**

#redis连接信息 pom中获取

jedis.pool.configstr=${jedis.pool.configstr}

jedis.pool.password=${jedis.pool.password}

jedis.pool.timeout=3000

jedis.pool.max-total=300

jedis.pool.max-idle=200

jedis.pool.max-wait-millis=1000

1. **swagger.properties配置文件信息**

#api作者

swagger.contact-name=Young

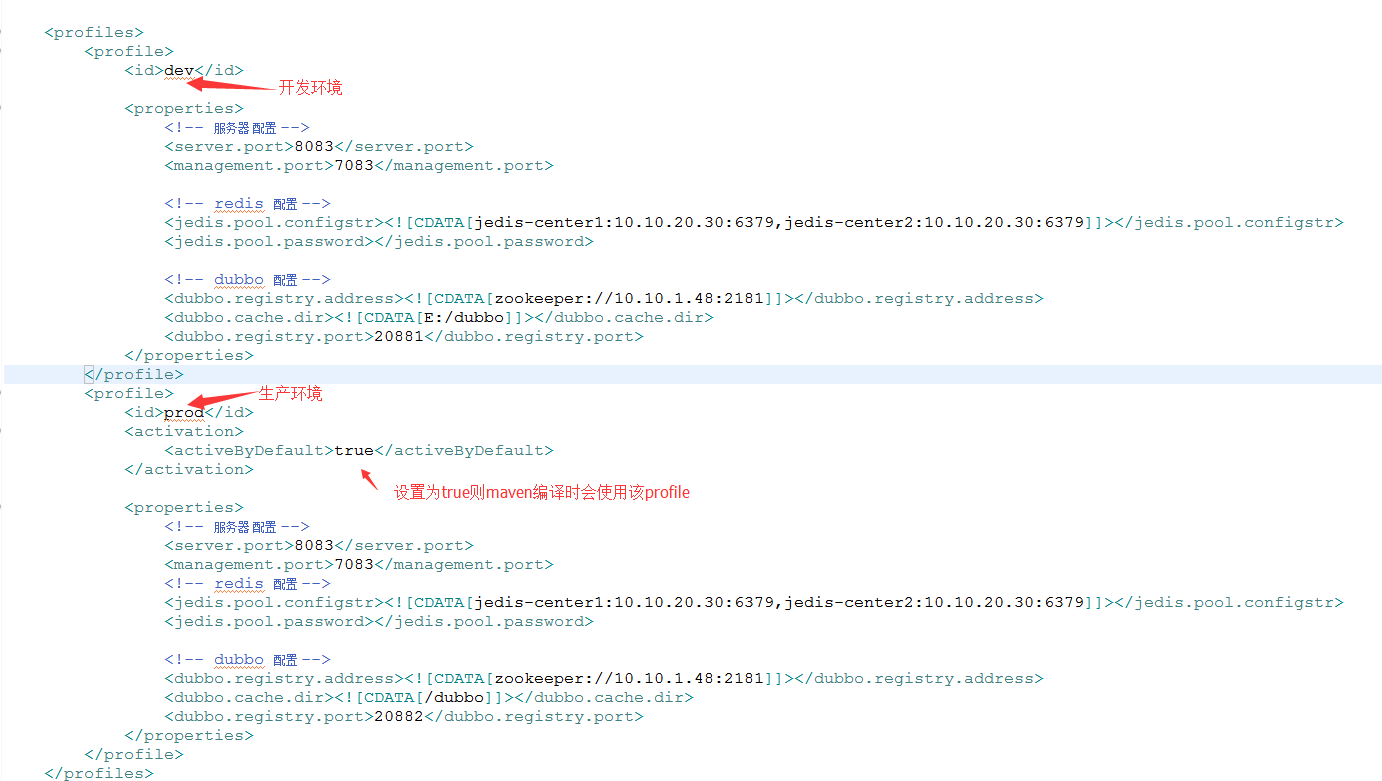
#api作者主页地址

swagger.contact-url=

#api作者联系地址

swagger.contact-email=zhiyong.yang@ubtrobot.com

## 6、pom文件根据不同的开发环境添加具体的配置信息

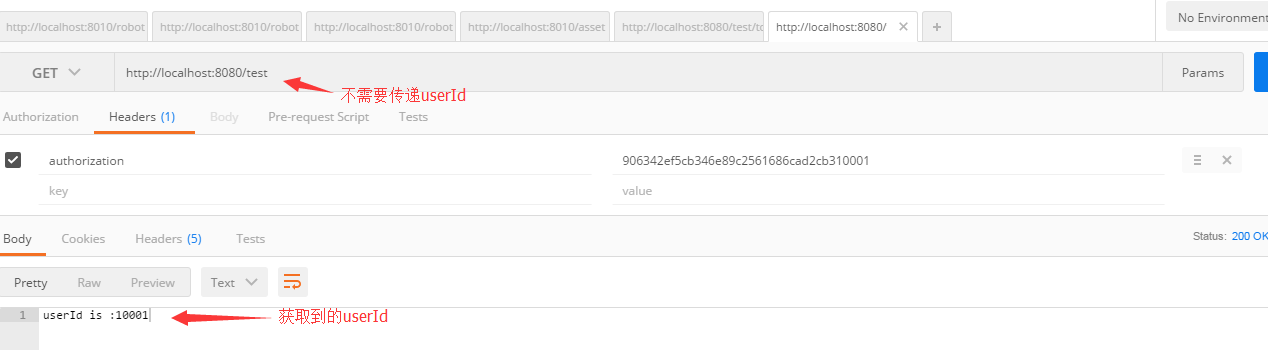


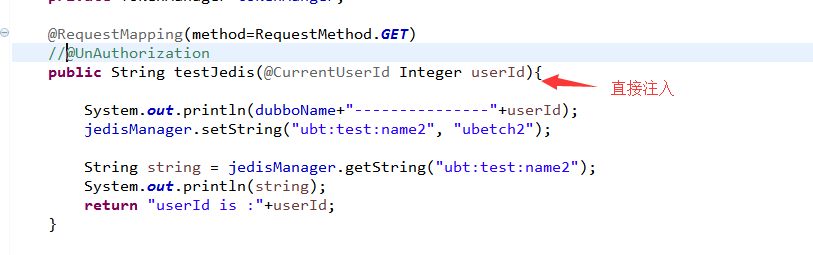
# **开发说明**

## 权限校验

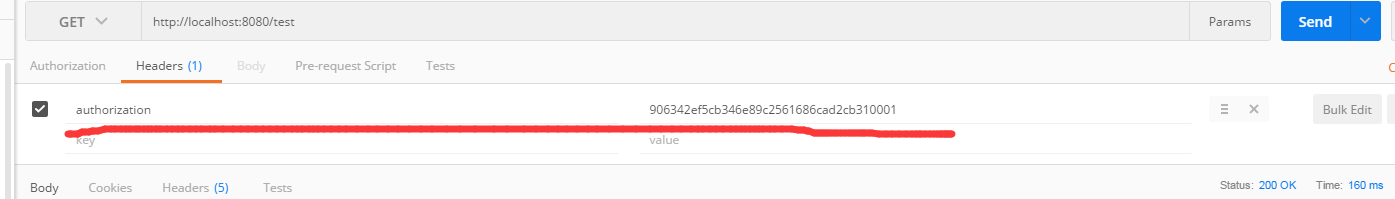
a、如果类中所有的接口都要验证token，在类上加上@Autorization，如果是方法需要token验证，则在方法上使用@Autorization注解即可

b、token 中已经包含了用户的id信息，接口开发时不需要再传入userId参数，如需使用userId，使用@CurrentUserId注入即可，如图所示：





c、所有的restful请求（get,post,put,delete等）token都放入请求的header中，如图所示



d、如果不进行token验证则不能使用@CurrentUserId标签注入userId

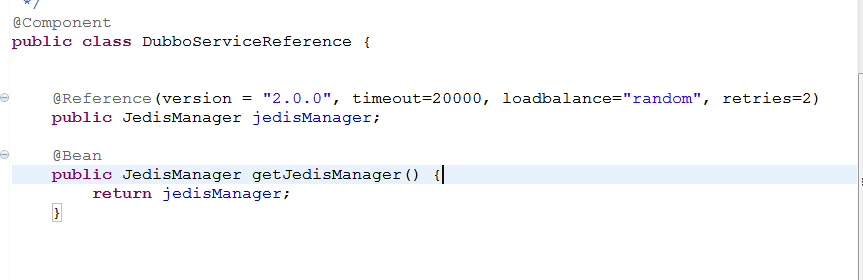
## **dubbo使用**

1. dubbo提供服务

@Service(protocol="dubbo",timeout=30000,version="2.0.0",loadbalance="random",retries=2,interfaceName="接口名称")

\*提供的服务类会生成动态代理（比如添加事物，缓存，日志等注解），需明确指定interfaceName

b、dubbo订阅服务



\*将订阅的服务注解为一个bean，其他类中就可以和spring bean一样进行自动注入

## **jedis使用**

@Autowired

**private** JedisManager jedisManager;

1. **cache使用**
2. 数据库数据缓存使用@DBCacheAnnotation注解

\*@DBCacheAnnotation注解只能在mapper包的类中使用，**缓存分页查询的数据会导致分页页码无法获取，所以请勿缓存分页数据**

**b、方法数据缓存使用**@Cache注解

\*由于缓存的方法数据有可能被其他的方法改变，所以务必设置一个过期时间如@Cache(expire=3600)

## **服务日志记录**

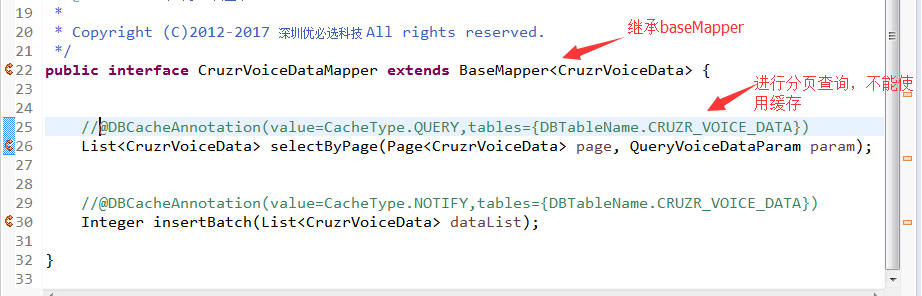
在需要写日志的服务方法上使用@Log(value=“方法功能”, type=LogType.QUERY)即可，其中value是说明方法作用，type是说明方法类型，详见枚举类LogType。

## **使用mybatis-plus开发流程**

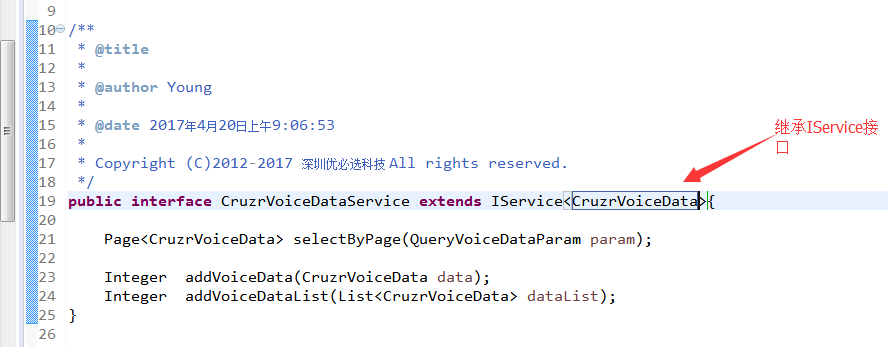
1. entity类中需明确指定类对应的表名和主键id，如图所示

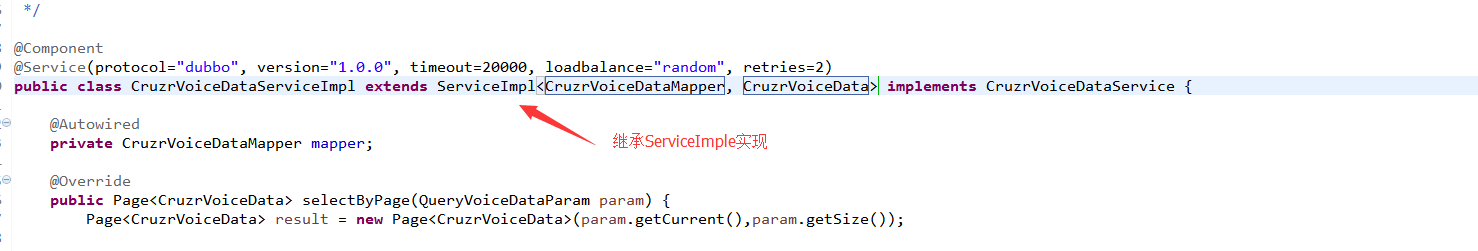


b、mapper类基础baseMapper



c、service 接口基础IService接口，Service实现继承ServiceImple实现，单表的增删改查操作基本上就不需要再自己写代码，直接使用对应的方法即可





d、自定义分页查询



