# 博客代码规范

## 一、命名规范

#### 1.包名永远用小写，不要用大写写包名

#### 2.dao连接数据库层使用以下命名

新增🡪insert

修改🡪update

删除🡪delete

查询🡪select

#### 3.controller和service层使用

新增🡪add

修改🡪edit

删除🡪remove

查询🡪find

#### 4.请求url格式不要有大写，如果嵌套需要就再写多一层

例如/user/add

**/apisuite/add** → **添加API接口**

**/apisuite/remove** → **删除API接口**

**/apisuite/all** → **查询API所有接口**

**/apisuite/detail** → **查询API接口详情**

**/apisuite/edit** → **修改API接口**

除了基本的增删查改，还会有一些特定的业务操作，也是同理：

**/apisuite/enable** → **启用API接口**

**/apisuite/disable** → **停用API接口**

#### 5.请求URL的方式

新增使用post方式

修改使用put方式

删除使用delete方式

查询使用get方式

#### 6.状态启动和停止使用两个固定方法写，命名如下

启用🡪enable

停用🡪 disable

#### 7.数据模型命名（co，dto，vo）

**CO,DTO,VO这几个字符都是用大写**

例如：

添加用户信息: UserAddCO

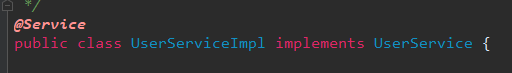
修改用户信息: UserEditCO

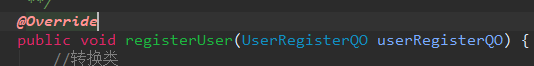
查询用户信息详情: UserInfoVO

查询用户信息列表: UserListVO

#### 8.类名首字符大写，驼峰式，方法名首字母小写，驼峰式

例如:





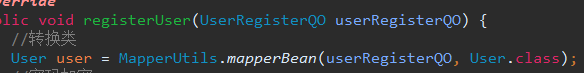
## 二、传参规范和数据模型使用

#### 1.Controller多个传输接收使用CO

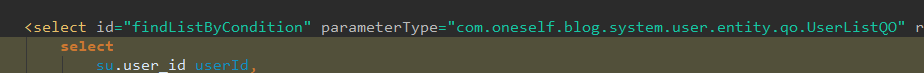
例如：



#### 2.Service新增和修改使用CO转pojo实体类

例如：

#### 3.查询直接用CO传到数据库

例如：

#### 4.数据库返回类使用VO接收

例如：

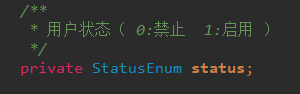


#### 5.如果service还需要处理DTO的参数，使用DTO转成VO返回

## 三、使用枚举和注释

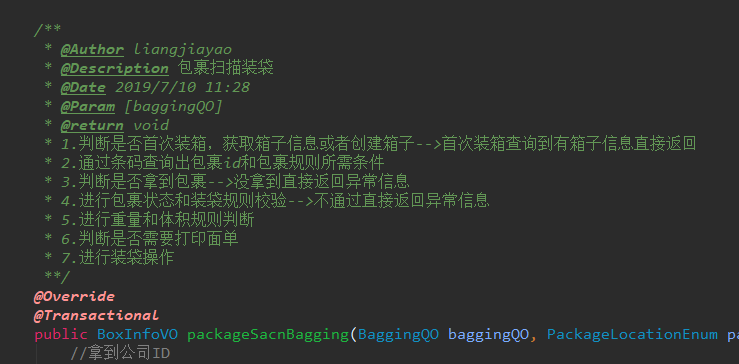
#### 1.数据库的状态值使用枚举类进行传参ENUM

例如：



#### 2.写好注释，逻辑复杂最好在方法上的注释写好步骤

例如：

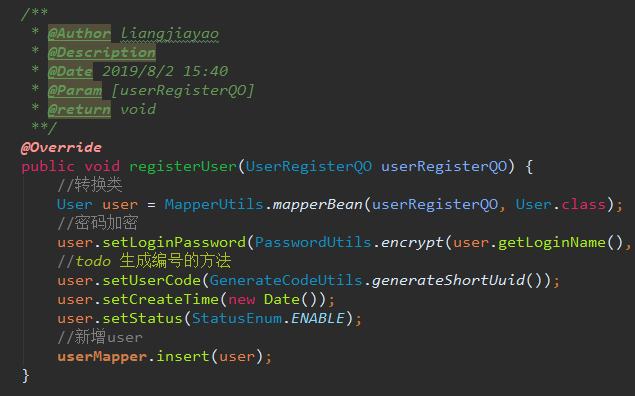


## 四、数据库和类的规范

#### 1.代码一个方法尽量不要超过80行，超过就抽逻辑出来另外写一个方法，主方法调用

#### 2.代码编写规范，层次感分明

例如：



#### 3.数据库写sql的时候按照格式规范写好，清晰明了

例如：**一定要层次感分明**

