# 命名规范

## 方法名、参数名、成员变量、局部变量都统一使用 lowerCamelCase 风格。

例： localValue / getHttpMessage() / inputUserId

## 常量命名全部大写，单词间用下划线隔开，力求语义表达完整清楚，不要嫌名字长。

例：MAX\_STOCK\_COUNT / CACHE\_EXPIRED\_TIME

## 包名统一使用小写，点分隔符之间有且仅有一个自然语义的英语单词。包名统一使用单数形式，但是类名如果有复数含义，类名可以使用复数形式。

例：应用工具类包名为 com.alibaba.ei.kunlun.aap.util、类名为 MessageUtils（此规则参考 spring 的框架结构）

## Service/DAO 层方法命名规约

1） 获取单个对象的方法用 get 做前缀。例：getUserDetail

2） 获取多个对象的方法用 list 做前缀，复数结尾，如：listObjects。

3） 获取统计值的方法用 count 做前缀。

4） 插入的方法用 save/insert/add/create 做前缀。

5） 删除的方法用 remove/delete 做前缀。

6） 修改的方法用 update 做前缀。

## 实体类领域模型命名规约

1） 数据对象：xxxDO，xxx 即为数据表名。

2） 数据传输对象【接收值】：xxxDTO，xxx 为业务领域相关的名称。

3） 展示对象【返回值】：xxxVO，xxx 一般为网页名称。

4） POJO 是 DO/DTO/BO/VO 的统称，禁止命名成 xxxPOJO。

## 禁止用is开头命名字段、属性名等

例：is\_delete【禁止】

## 接口命名规范

接口名称定义遵循数据库curd【创建（Create）、更新（Update）、读取（Retrieve）和删除（Delete）】

使用 {模块名}/{版本号}/{功能}。

例：http://abc.com/im-commo-category/v1/create

接口名称定义:

1. 查询单个 get

2. 查询多个 list

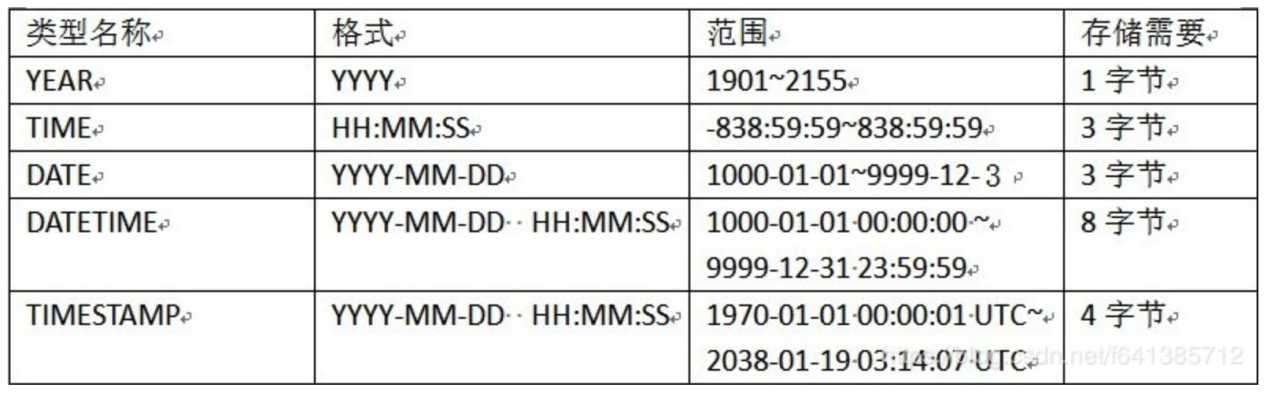
3. 添加 create

4. 更新 update

5. 删除delete

## 每张表必须有create\_time，update\_time，create\_by字段

create\_time创建时间，update\_time更新时间，为服务器时间，使用datetime/timestamp数据类型。



# 注意事项

## BigDecimal 的等值比较应使用 compareTo()方法，而不是 equals()方法。

说明：equals()方法会比较值和精度（1.0 与 1.00 返回结果为 false），而 compareTo()则会忽略精度。

## 判断字符串是否等于某个值的写法

例：if（“value”.equals(user.getUserName())）{

}

禁止：if（user.getUserName().equals(“value”)）{

}

## 所有增删改的操作必须加事务

## 所有字段名、表名、属性名、方法名、类名必须加注释

## 在 if/else/for/while/do 语句中必须使用大括号。

即使只有一行代码，也禁止不采用大括号的编码方式：if (condition) statements;

## 前后端数据列表相关的接口返回，如果为空，则返回空数组[]或空集合{}。

## finally 块必须对资源对象、流对象进行关闭，有异常也要做 try-catch。

## 生产环境禁止直接使用 System.out 或 System.err 输出日志或使用e.printStackTrace()打印异常堆栈。



# 分支管理

master分支：主干只用于其他分支合并

release-20180611分支：上线使用

develop分支：开发分支，feature分支合并及test分支修改后合并

test分支：专门给测试人员进行测试的分支，test-xxx子分支修改bug后合并

test-20180611：测试提bug后，在此分支上修改

feature-xxx分支：多个开发分支，按模块新建分支，开发完成合并到develop上。

# 补充

## 表主键

主键id类型为char32位小写

## 乐观锁

数据库更新操作需要加乐观锁

## 参数校验

validation框架，常规校验必须添加，避免报错