# 开发规范

## 前言

一个项目的健壮性、可用性、可维护性，不在于使用最新的、最流行的框架，而在于是否从一而终保持代码的整洁；在于是否从一而终的保持代码及时重构；在于是否从一而终的保证代码的可读性。因此，为了保证代码的整洁，保证代码重构的质量，保证代码的持续可读性。特制订用于本项目的开发规范。本规范将涉及Java语言、数据库语言、JavaScript语言、CSS样式语言、HTML语言等语言的规范化（若以后会有其他语言加入到该项目，则进一步拓展）。

本项目的Java开发是基于IDEA进行的，因此Java规范（包括JSP）会与IDEA结合。而JavaScript、CSS、HTML会在sublime编辑器上进行，因此相关规范会与sublime关系较为紧密。数据库则统一规范化。

## Java规范

* 命名规范
* 代码中禁止出现无意义的名称（例：a1,a2…），任意一个位置出现的名称都应该有其明确的意义。
* 代码中出现的名称应该尽可能的含义准确，避免使用多义词。不要使用不是众所周知的缩写。明确是王道
* 避免使用编码。Java语言是强类型的，因此不要在命名中出现类型含义。禁止使用成员前缀。不要用前缀标出作用域，编辑器会告诉你使用的变量的作用域。接口类不要使用I开头，如果一定要添加的话，在实现类的末尾使用imp。
* 一律遵守驼峰式命名规则。

包名首字母小写。

类名首字母小写。类名应该是名字或名字短语，不应该是动词

方法名首字母小写。方法名应该是动词或动词短语。

静态final域采用全部大写字母，单词间使用下划线（\_）分割

* 函数
* 函数应该尽可能的短小。函数的缩进层级不应该多于一层或两层。
* 函数应该只做一件事。判断函数是否不知不止做了一件事，可以通过太函数是否能够再拆分出一个函数。而且新函数不仅是单纯地全是其实现，只做一件事的函数无法被合理地切分为多个区段
* 每个函数一个抽象层级
* 使用描述性的名称。这里有一个原则：如果每个例程都让你感到深合己意，那就是整洁代码。
* 参数个数要尽可能保持在三个以下，若超出三个以上，要考虑封装参数了
* 函数要么做什么事，要么回答什么事，不允许一个函数出现两者兼顾的情况
* Don’t repeat yourself! Don’t repeat yourself! Don’t repeat yourself!重要的事情说三遍！必须保证DRY原则
* 注释
* 唯一真正好的注释是想办法不去写注释
* 注释用来通基本信息
* 注释用来对意图进行解释
* 注释用来阐释一些比较晦涩的参数或返回值
* 注释用来警告其他程序员会出现某种后果
* TODO注释是一种程序员认为应该做，但由于某些原因还没有做的工作。
* 注释可以用来放大某种看来不合理之物的重要性。
* 公共API一定要提供良好的Javadoc
* 格式
  + - 所有的变量声明应该集中在类的开始
    - 变量声明结束和方法声明开始之间应间隔一行
    - 不同方法之间应该间隔一行
    - 代码块之间的缩进设置为4个空格符。其他参考统一的格式化文件
* 测试
  + - 保证每一个逻辑代码被测试覆盖。
    - 在编写不能通过的单元测试前，不可编写生产代码
    - 只可编写刚好无法通过的单元测试，不能编译也算不通过
    - 只可编写刚好足以通过当前失败测试的生产代码
    - 测试应该够快
    - 测试应该相互独立
    - 测试可以任何环境中重复通过
    - 测试应该有布尔值输出
    - 测试应及时编写

## 数据库规范

命名规范

数据表命名规范：

1. 采用26个英文字母(区分大小写)和0-9的自然数(经常不需要)加上下划线'\_'组成，命名简洁明确，多个单词用下划线'\_'分隔
2. 全部小写命名，禁止出现大写
3. 禁止使用数据库关键字，如：name，time ，datetime，password等
4. 表名称不应该取得太长（一般不超过三个英文单词）
5. 表的名称一般使用名词或者动宾短语
6. 用单数形式表示名称，例如，使用 employee，而不是 employees 字段名：除特殊含义的单词外，均使用小写字母，不同单词之间使用\_分割
7. 表必须填写描述信息（备注）
8. 数据表必须有主键，且建议均使用auto\_increment的id作为主键（与业务无关）,和业务相关的要作为唯一索引
9. 默认使用utf8字符集

字段命名规范：

1. 采用26个英文字母(区分大小写)和0-9的自然数(经常不需要)加上下划线'\_'组成，命名简洁明确，多个单词用下划线'\_'分隔
2. 全部小写命名，禁止出现大写
3. 字段必须填写描述信息
4. 禁止使用数据库关键字
5. 字段名称一般采用名词或动宾短语
6. 采用字段的名称必须是易于理解，一般不超过三个英文单词
7. 在命名表的列时，不要重复表的名称
8. 不要在列的名称中包含数据类型
9. 字段命名使用完整名称，禁止缩写
10. 具备统一前缀，对相关功能的表应当使用相同前缀，如acl\_xxx，house\_xxx,ppc\_xxx；其中前缀通常为这个表的模块或依赖主实体对象的名字，通常来讲表名为：业务\_动作\_类型，或是业务\_类型；

字段类型规范：

1. 所有字段在设计时，除以下数据类型timestamp、image、datetime、smalldatetime、uniqueidentifier、binary、sql\_variant、binary 、varbinary外，必须有默认值，字符型的默认值为一个空字符值串’’，数值型的默认值为数值0，逻辑型的默认值为数值0
2. 系统中所有逻辑型中数值0表示为“假”，数值1表示为“真”，datetime、smalldatetime类型的字段没有默认值，必须为NULL
3. 用尽量少的存储空间来存储一个字段的数据
4. 用合适的字段类型节约空间

字符转化为数字(能转化的最好转化，同样节约空间、提高查询性能)

避免使用NULL字段(NULL字段很难查询优化、NULL字段的索引需要额外空间、NULL字段的复合索引无效)

少用text类型(尽量使用varchar代替text字段)

大小写规范：

1. 所有关键字必须大写，如：INSERT、UPDATE、DELETE、SELECT及其子句，IF……ELSE、CASE、DECLARE等
2. 所有函数及其参数中除用户变量以外的部分必须大写
3. 在定义变量时用到的数据类型必须小写

常用表名约定：

1. 说明：表前缀用项目名称首字母缩写；所以表名都小写，单词之间用下划线分开，单词都用单数形式
2. user – 用户
3. category – 分类
4. goods – 商品、产品等一切可交易网站的物品都用此命名
5. good\_gallery – 物品的相册
6. good\_cate – 物品的分类，除了单独作为表名，其他地方分类单词一律用缩写cate
7. attr – 属性
8. article – 文章、新闻、帮助中心等以文章形式出现的，一般都用此命名
9. cart – 购物车
10. feedback – 用户反馈
11. order – 订单
12. site\_nav – 包括页头和页尾导航
13. site\_config – 系统配置表
14. admin – 后台用户 【RBAC标准表】
15. role – 后台用户角色【RBAC标准表】
16. access – 后台操作权限，相当于action【RBAC标准表】
17. role\_admin – 后台用户对应的角色【RBAC标准表】
18. access\_role – 后台角色对应的权限【RBAC标准表】