# 贝如编码规范

## 服务端

1.1.1. 【强制】代码中的命名均不能以下划线或美元符号开始，也不能以下划线或美元符号结束。

反例：\_name/\_\_name/$Object/name\_/name$/Object$

1.1.2. 【强制】代码中的命名严禁使用拼音与英文混合的方式，更不允许直接使用中文的方式。

说明：正确的英文拼写和语法可以让阅读者易于理解，避免歧义。注意，即使纯拼音命名方式也要避免采用。

正例：alibaba/taobao/youku/hangzhou 等国际通用的名称，可视同英文。

反例：DaZhePromotion [打折] / getPingfenByName() [评分] / int 某变量 = 3

1.1.3. 【强制】类名使用UpperCamelCase风格，必须遵从驼峰形式

正例：MarcoPolo/XmlService/TcpUdpDeal/TaPromotion

反例：macroPolo/XMLService/TCPUDPDeal/TAPromotion

1.1.4. 【强制】方法名、参数名、成员变量、局部变量都统一使用lowerCamelCase风格，必须遵从驼峰形式。

正例： localValue/getHttpMessage()/inputUserId

1.1.5. 【强制】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。

正例：MAX\_STOCK\_COUNT

反例：MAX\_COUNT

1.1.6. 【强制】抽象类命名使用Abstract或Base开头；异常类命名使用Exception结尾；测试类命名以它要测试的类的名称开始，以Test结尾。

1.1.7. 【强制】中括号是数组类型的一部分，数组定义如下：String[] args;

反例：使用String args[]的方式来定义。

1.1.8. 【强制】POJO类中布尔类型的变量，都不要加is，否则部分框架解析会引起序列化错误。

反例：定义为基本数据类型Boolean isDeleted；的属性，它的方法也是isDeleted()，RPC框架在反向解析的时候，“以为”对应的属性名称是deleted，导致属性获取不到，进而抛出异常。

1.1.9. 【强制】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。

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

1.1.10. 【强制】杜绝完全不规范的缩写，避免望文不知义。

反例：AbstractClass“缩写”命名成AbsClass；condition“缩写”命名成 condi，此类随

意缩写严重降低了代码的可阅读性。

1.1.11. 【推荐】为了达到代码自解释的目标，任何自定义编程元素在命名时，使用尽量完整的单词组合来表达其意。

正例：从远程仓库拉取代码的类命名为 PullCodeFromRemoteRepository。

反例：变量 int a; 的随意命名方式。

1.1.12. 【推荐】如果模块、接口、类、方法使用了设计模式，在命名时体现出具体模式。

说明：将设计模式体现在名字中，有利于阅读者快速理解架构设计理念。

正例：public class OrderFactory;

public class LoginProxy;

public class ResourceObserver;

1.1.13. 【推荐】接口类中的方法和属性不要加任何修饰符号（public 也不要加），保持代码的简洁

性，并加上有效的Javadoc注释。尽量不要在接口里定义变量，如果一定要定义变量，肯定是

与接口方法相关，并且是整个应用的基础常量。

正例：接口方法签名：void f();

接口基础常量表示：String COMPANY = "alibaba";

反例：接口方法定义：public abstract void f();

说明：JDK8 中接口允许有默认实现，那么这个default方法，是对所有实现类都有价值的默认实现。

1.1.14. 接口和实现类的命名有两套规则：

1）【强制】对于Service和DAO类，基于SOA的理念，暴露出来的服务一定是接口，内部

的实现类用Impl的后缀与接口区别。

正例：CacheServiceImpl实现CacheService接口。

2） 【推荐】 如果是形容能力的接口名称， 取对应的形容词做接口名（通常是–able的形式）。

正例：AbstractTranslator实现 Translatable。

1.1.15. 【参考】枚举类名建议带上Enum后缀，枚举成员名称需要全大写，单词间用下划线隔开。

说明：枚举其实就是特殊的常量类，且构造方法被默认强制是私有。

正例：枚举名字为ProcessStatusEnum 的成员名称：SUCCESS / UNKOWN\_REASON。

1.1.16. 【参考】各层命名规约：

A) Service/DAO层方法命名规约

1） 获取单个对象的方法用get/find做前缀。

2） 获取多个对象的方法用list/find做前缀。

2） 分页获取数据的方法用get/find做前缀，使用Page做后缀。

3） 获取统计值的方法用count做前缀。

4） 插入的方法用save/insert做前缀。

5） 删除的方法用remove/delete做前缀。

6） 修改的方法用save/update做前缀。

B) 领域模型命名规约

1） 数据对象：xxxDo，xxx为对应业务用途。

2） 数据传输对象：xxxDTO，xxx为业务领域相关的名称。

3） 接口响应对象：xxxResult，xxx一般对应接口名称。

4） 接口接收参数：xxxParam，xxx一般对应接口名称。

## 管理系统前端

* + 1. **css样式**

1.2.1.1 类名命名以"-"连接或者驼峰命名法，尽量不得超过4个单词，符合语义化，可简写（简写需注释说明）

正确示范：.dialog-header-title .table-body-row .input-psw .dialogHeaderTitle

1.2.1.2 命名保持单个单词，符合语义化

正确示范：#seller

错误示范：#t1

1.2.1.3 不使用过分嵌套定义样式（除特定封装组件内部可以嵌套定义样式）

正确示范：.table{} .table-header{}

错误示范：.dialog p span{}

1.2.1.4 采用多个类的组合去实现，通过定义一些基础类，尽量避免相似样式，重复定义，不相似可自行定义对应类名去实现

示范：.flex-row-center{align-items: center}

1.2.1.5 类名命名基于标签结构去命名，而不是业务

正确示范：.input-container .input-title .input-content .dialog-header .modal-content .modal-footer .row-list

错误示范：.buy-box .company-list .client-dialog

1.2.1.6 如果可以的情况下，尽量避免使用内联样式

* + 1. **JavaScript样式**

1.2.2.1 变量，函数采用驼峰命名

示范：let searchBar = '';let getListData = ()=>{}

1.2.2.2 类名采用首字母大写命名

示范：class PriceCalculator {}

1.2.2.3 常量命名采用全大写命名，用'\_'连接

示范：const DEFAULT\_TIMES = 20

1.2.2.4 变量、常量、类名均采用名词，不建议使用动词以及不达意单词，需一眼能明白该变量的作用

正确示范：let dialogTitle = '查看单位';

const INPUT\_TYPE = 'select';

class CompanySelector {};

错误示范：let num1 = 1;const T1 = 't1';

1.2.2.5 方法/函数定义采用动词+名词的形式，需要能直接表示该方法的作用

正确示范：let getSelectOptions = ()=>{};

let editCompanyData = ()=>{};

let submitResult = ()=>{};

错误示范：let index =()=>{};

1.2.2.6 如果多个方法内均使用同一段代码段，建议抽出来单独定义，再调用