微链代码规范

|  |  |  |
| --- | --- | --- |
| 版本 | 作者 | 说明 |
| 1.0 | 解雷 |  |
|  |  |  |

# 通用

## 代码工具

### IDE

必须使用IDEA-CE

### 格式化代码

必须使用模式格式化模板,代码编写完必须格式化

## 命名

类名,方法名,变量名(除静态常量),参数名使用驼峰规则

类名首字母大写

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。

静态常量(包含枚举)使用全大写加下划线规则

Boolean类型变量不得使用is前缀

例:

**public class** RedisService {

**private boolean *success*** = true;

**private final static int *VIEW\_COUNT*** = 1;

**private final static** Logger ***LOG*** = Logger.*newInstance*(RedisService.**class**);

**private** StringRedisTemplate **stringRedisTemplate**; **public void** putString(String key, String value) {  
 checkStringKey(key);  
 **stringRedisTemplate**.boundValueOps(key).set(value);  
}

}

## 注释

### 类注释

类作用说明

### 方法注释

方法作用说明

参数说明

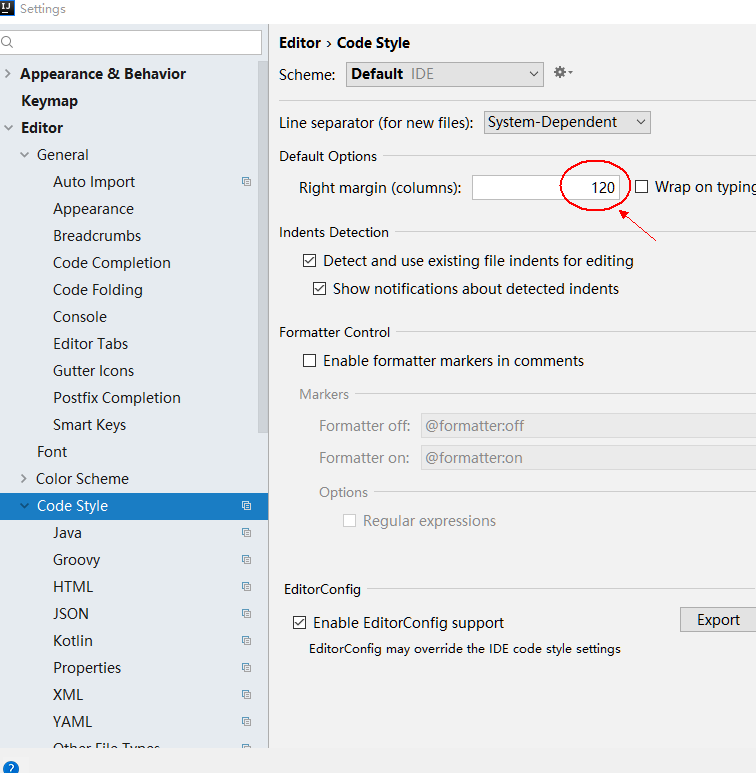
### 成员变量注释

成员变量作用说明

### 代码逻辑注释

负责逻辑详细描述关键性节点作用

## 每行字数限制120



## 代码行数

### 类文件

每个Java Class类不得超过1000行代码,多于1000行必须拆分

### 方法

每个方法体不等超过100行代码,超过100行必须拆分

### 参数

方法参数个数不得超过5个,超过5个必须使用Bean承载

### 变量

变量名必须不得超过30个字符

## 编码风格

### 接口

接口定义原则为按需定义,一下两种情况建议定义接口

1. 已知需要多种实现方式
2. 未知实现方式,仅作为参数传递

强制不能使用接口的情况

Service层不得使用接口,除非满足上面的要求

### 抽象类

公共方法尽量放在抽象类中

### 国际化

除注释和日志外,代码中不能包含中文

所有文案类文本必须使用国际化方式处理

### Bean和POJO

Bean:用于对外暴露参数,属性为public,原则上不得含有get/set方法

POJO:用户ORM,必须包含get/set方法

### h代码层级关系

#### 三层结构

分为Controller Service Mapper三层

Controller层只能引用Service层实例

Service层只能引用Service或Mapper或Client层实例

#### 四层结构

分为Controller Service Manager Mapper三层

Controller层只能引用Service层实例

Service层只能引用Service或Manager或Client层实例

Manager层只能引用Manager或Mapper或Client层实例

### 返回状态码

除业务需要特殊定义外必须使用com.welian.code.ResultCodes.Code中的状态码

### API

使用RESTful风格

#### 查询

##### 简单条件查询

条件1:

1. 只有一个参数
2. 参数值只包含数字和英文字符
3. 最多返回一条数据

使用GET方式,必须使用[select/名称]形式定义,名词必须使用单数

[http://api.welian.com/select/user/{id}](http://api.welian.com/select/user/%7bid%7d)

[http://api.welian.com/select/user/phone/{phone}](http://api.welian.com/select/user/phone/%7bphone%7d)

条件2:

1. 包含多个参数
2. 参数之间有关联关系
3. 参数值只包含数字和英文字符
4. 最多返回一条数据

使用GET方式,必须使用[select/名称]形式定义,名词必须使用单数

[http://api.welian.com/select/org/{orgId}user/{uid}](http://api.welian.com/select/org/%7borgId%7duser/%7buid%7d)

[http://api.welian.com/select/org/{orgId}/phone/{phone}](http://api.welian.com/select/org/%7borgId%7d/phone/%7bphone%7d)

##### 复杂条件查询

条件1:

1. 参数值可能包含中文
2. 可返回多条数据

使用POST方式, 必须使用[select/名称]形式定义,名词必须使用复数

<http://api.welian.com/select/users>

Body: JSON格式(包含条件)

#### 新增

条件:

1. 数据源中不存在
2. 若存在抛出异常(Code .COMMON\_ERROR\_EXIST)

使用POST方式,api中只能包含名称,名词必须使用复数

<http://api.welian.com/users>

Body: JSON格式

#### 修改

##### 修改全部

条件1:

1. 已知数据id等唯一条件
2. 修改数据全部属性

使用PUT方式,api中只能包含名称,名词必须使用单数

[http://api.welian.com/user/{id}](http://api.welian.com/user/%7bid%7d)

[http://api.welian.com/user/{orgId}/{id}](http://api.welian.com/user/%7borgId%7d/%7bid%7d)

Body: JSON格式

条件2:

1. 已知数据定位条件
2. 修改全部属性

使用PUT方式,api中只能包含名称,名词必须使用复数

http://api.welian.com/users

Body: JSON格式(包含条件)

##### 修改部分

条件1:

1. 已知数据id等唯一条件
2. 修改数据全部属性

使用PATCH方式,api中只能包含名称,名词必须使用单数

[http://api.welian.com/user/{id}](http://api.welian.com/user/%7bid%7d)

[http://api.welian.com/user/{orgId}/{id}](http://api.welian.com/user/%7borgId%7d/%7bid%7d)

Body: JSON格式

条件2:

1. 已知数据定位条件
2. 修改全部属性

使用PATCH方式,api中只能包含名称,名词必须使用复数

http://api.welian.com/users

Body: JSON格式(包含条件)

#### 删除

条件1:

1. 已知数据id等唯一条件
2. 修改数据全部属性

使用DELETE方式,api中只能包含名称,名词必须使用单数

[http://api.welian.com/user/{id}](http://api.welian.com/user/%7bid%7d)

[http://api.welian.com/user/{orgId}/{id}](http://api.welian.com/user/%7borgId%7d/%7bid%7d)

Body: JSON格式

条件2:

1. 已知数据定位条件
2. 修改全部属性

使用DELETE方式,api中只能包含名称,名词必须使用复数

http://api.welian.com/users

Body: JSON格式(包含条件)

## 依赖

线上服务不能依赖SNAPSHOT版本

## 版本号

主版本号.次版本号.修订号

1. 主版本号：当做了不兼容的 API 修改，或者做了重大调整。
2. 次版本号：普通正常迭代。
3. 修订号：修复 bug或临时插入的迭代。

起始版本号必须为： 1.0.0 ，每个版本号范围在0~99，最大版本号为99.99.99

## 日志

必须使用com.welian.util.Logger输出日志

不能使用System.out.print\*(\*)输出日志

## 测试用例

包含逻辑的代码必须有测用例

举例:

controller层

service层

mapper中自定义sql

工具类等等

# 包

com.welian

com.welian.controller

com.welian.service

com.welian.mapper

com.welian.config 配置相关

com.welian.init 启动加载

com.welian.lanuch 启动类

com.welian.bean

com.welian.pojo

com.welian.client

# 类

## 子类

类中只能包含本类,接口,枚举以及非public子类

违反说明:

public class A {

public class B {  
}

}

# 方法

1. 返回值定义为基本类型时,return语句必须返回基本类型
2. 参数定义基本类型时,调用方必须确保Not Null

# 编码注意点

## 基本类库

Json处理: com.welian.util.JSONUtil

时间处理: joda-time和com.welian.util.DateUtil

MD5: com.welian.util.MD5

Base64: com.welian.util.Base64

加解密: com.welian.util. CipherUtil

缓存: com.welian.cache

文件: com.welian.util.FileUtil

## 类型转换

1. 定义为基本类型的变量,在不明确值的情况下不得进行强制转换

反例: int count = JsonObject.get(“count”);

## 控制语句

1. if where for switch等控制语句必须使用{}

## 集合类型

1. List,Map使用时必须指明泛型类型,

没有特殊要求情况下,List使用ArrayList ;Map 使用 HashMap

## 并发

1. 必须使用com.welian.pool.CommonExecutorPool工具类