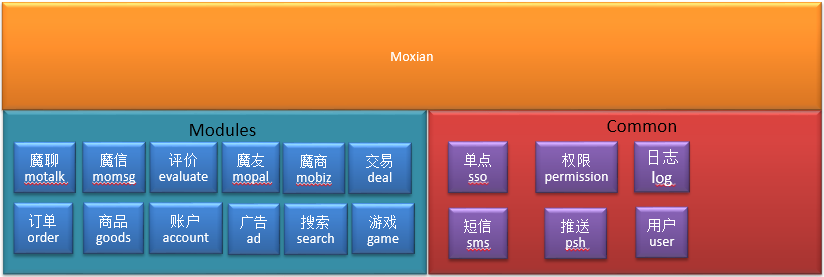
Java开发规范

# 项目构建结构



# 代码命名规范

* 英文命名

所有名称统一使用英文命名，不许使用拼音。

* 包命名

包名按照域名的范围从大到小逐步列出由一组以“.”连接的标识符构成，通常第一个标识符为符合网络域名的两个或者三个英文小写字母，com.moxian.子系统名.模块名.层.xxx如com.moxian.mobiz.mechantsmanagement.xxxx。

* 类，接口命名

类的名字必须由大写字母开头而单词中的其他字母均为小写;如果类名称由多个单词组成，则每个单词的首字母均应为大写例如TestPage;如果类名称中包含单词缩写，则这个所写词的每个字母均应大写，如：XMLExample,还有一点命名技巧就是由于类是设计用来代表对象的，所以在命名类时应尽量选择名词。

* 方法名

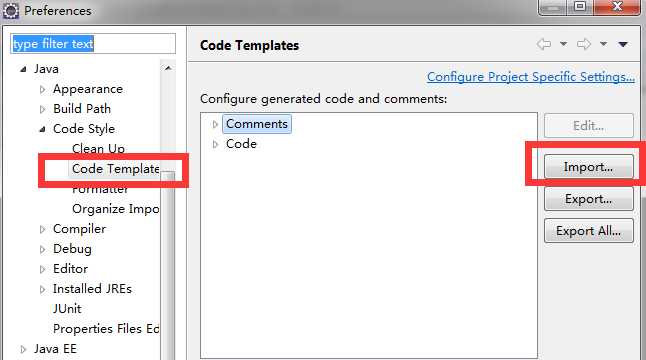
方法的名字的第一个单词应以小写字母作为开头，后面的单词则用大写字母开头。可以为动词或动词+名词组合。

* 变量名

普通变量命名应该采用首字母小写，其他字母首字母大写的方式，final static变量的名字应该都大写，并且指出完整含义。如果一个常量名称由多个单词组成，则应该用下划线来分割这些单词。

# 代码注释

每个类、接口、方法中，必须加上注释。进入Eclipse->Windows->Preference，导入注释模板，导入后注意把个人信息添加上





# 接口返回规范

接口返回统一使用返回json对象的方式，result、code、msg为必须返回属性，data中再封装业务的json对象。

result, result=true取得数据，result=false未取得数据

code, 8位数字状态码 具体的状态码详细见统一状态码说明表格。

msg, 返回信息

**例子：**

**接口说明：获取魔商首页广告数据**

**返回说明：**

|  |  |  |
| --- | --- | --- |
| **字段名** | **类型** | **说明** |
| result | bool | result=true取得数据，result=false未取得数据 |
| code | int | 状态码 详细见统一状态码说明 |
| msg | String | 返回信息 |
| data | Json数据类型 | 数据字段，json格式 |
| listData | Json数据类型 | 列表数据 |
| picUrl | String | 广告图片下载地址 |
| link | String | 点击广告跳转地址 |
| type | int | 0-魔信 1-魔抢 2-魔拍 3-优惠券 4-印花 |
| id | String | type为0时，该字段有效，魔信id |

正确的json返回结果

{

"result": true,

"code": 0,

"msg": "success",

"data": {

"listData": [{

" picUrl ": " http:\/\/rs.dev.moxian.com\/upload\/user\/285448\/1404987.jpg",

" link": "http://www.moxian.com/mogiftcard/mobileweb/mo\_voucher? ",

" type": 3,

" id ": 0

},

{

" picUrl ": " http:\/\/rs.dev.moxian.com\/upload\/user\/285448\/1404987.jpg",

" link": "http://www.moxian.com/mogiftcard/mobileweb/mo\_voucher? ",

" type": 4,

" id ": 0

},

{

" picUrl ": " http:\/\/rs.dev.moxian.com\/upload\/user\/285448\/1404987.jpg",

" link": "http://www.moxian.com/mogiftcard/mobileweb/mo\_voucher? ",

" type": 0,

" id ": 1233455

}]

}

}

错误的json返回结果

{

"result": false,

"code":10121021,

"msg": "failed!"

}

# 代码开发流程

# Rest API URL规范

## 接口命名规则

<http://ip:端口/v2/接口名>

IP：服务器IP地址

端口：Restful端口号

V2:版本号(2)

接口名：

命名规则：现有接口方法去第一个单词后，全小写命名，如：

用户余额信息实时查询能力，原接口名： queryBalanceRealSingle(String msisdn)

Restful接口:http://ip:端口/v2/balancerealsingle

## 参数规则

参数提交方式：application/www-form-urlencoded

参数命名：单词采取小写，复合词采取下划线分开的全小写命名。

参数规则：批量查询需有page\_size以及page\_num参数，避免一次性查询，部分参数需有默认值设定。

## 其它

* 使用 POST 方法在服务器上创建资源
* 使用 GET 方法从服务器检索某个资源或者资源集合
* 使用 PUT 方法对服务器的现有资源进行更新
* 使用 DELETE 方法删除服务器的某个资源

1. 资源有层次关系的时候 URI 模式的设计用复合的 URI 来帮助开发者更好的理解和设计资源

比如， 针对一个文档的评论， 他的 URI 模式可以设计成如下：

/ 文件夹 / [ 文件夹名 ] / 文件 / [ 文件名 ] / 评论 / [ 评论唯一标示 ]。

1. 使用过滤条件来帮助用户更准确地获取数据

最常见的过滤条件，是通过 URL 参数实现，

比如， / 环境工程系 / 学生 ? 籍贯 = 北京 & 性别 = 女。

可以使用正则表达式或者服务器可以理解的语法，

比如 ，/ 环境工程系 / 学生 ?filter=age between (15, 18)

POST 方法不建议使用

1. 排序

通常有 2 个参数，一个是用来排序的字段，一个是排序的升序降续方式。

比如我们可以用支持这样的参数组合的手段，提供基本的排序能力：?sortOrder=asc&sortField=age

1. 使用分页来帮助客户端处理大量数据

通常用一个以上参数来制定一个返回结果的区域，比较常见的有下面两种：

一种常见于用固定行数的表格来展示数据，用当前处于第几页和每页返回多少行数据来确定需要的数据，

比如，/ 所有学生 ?page=5&pagesize=50

另外一种常见于用更加灵活的界面展示数据，用从第几行开始，一共返回多少行数据来确定需要的数据，

比如，/ 所有学生 ?startIndex=27&count=22

1. 特殊资源类型

在 REST 这个以名词为核心的构架结构中，当你遇到一些动词特性比较强的操作，

而又很难用原始资源的增删改查来匹配的时候，不妨换个思路， 通过引入新的逻辑资源集合的方式， 来进行 API 的设计与规划

1. @QueryParam和@PathParam比较

A、 先来看@queryparam

1. @Path("/users")

2. public class UserService {

3.

4. @GET

5. @Path("/query")

6. public Response getUsers(

7. @QueryParam("from") int from,

8. @QueryParam("to") int to,

9. @QueryParam("orderBy") List<String> orderBy) {

10.

11. return Response

12. .status(200)

13. .entity("getUsers is called, from : " + from + ", to : " + to

14. + ", orderBy" + orderBy.toString()).build();

15.

16. }

17.

18. }

URL输入为:users/query?from=100&to=200&orderBy=age&orderBy=name

此时,输出为:

getUsers is called, from : 100, to : 200, orderBy[age, name]

要注意的是,跟@pathparam不同,@queryparam中,指定的是URL中的参数是以键值对的形式出现的,而在程序中

@QueryParam("from") int from则读出URL中from的值, 而@pathparem中,URL中只出现参数的值,不出现键值对,比如: “/users/2011/06/30”

B、@PathParam例子

1. @GET

2. @Path ("{year}/{month}/{day}")

3. public Response getUserHistory(

4. @PathParam("year") int year,

5. @PathParam("month") int month,

6. @PathParam("day") int day) {

7.

8. String date = year + "/" + month + "/" + day;

9.

10. return Response.status(200)

11. .entity("getUserHistory is called, year/month/day : " + date)

12. .build();

13.

14. }

# 7. Rest API Java开发注意事项

1. 新建资源或删除资源等，请求需要返回当前资源Id

2. Resource中的***BASE\_PATH*** 变量，统一使用全小写字母，且不能使用下划线“\_”

# 8. Git 基本使用流程

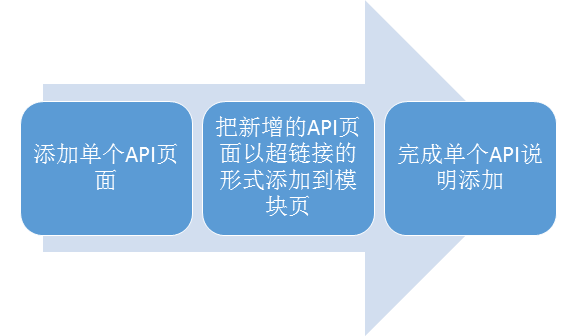


# 9. Wiki 接口说明添加规范

每个Rest API 接口需要在wiki上添加一个说明单页，注意添加到API所属的模块中。

Wiki地址：<http://wiki2.moxian.com/>

添加基本流程：



示例：

单个API页面：<http://wiki2.moxian.com/index.php?title=%E6%B7%BB%E5%8A%A0%E6%94%B6%E8%B4%A7%E5%9C%B0%E5%9D%80>

模块页：<http://wiki2.moxian.com/index.php?title=%E5%88%86%E7%B1%BB:MX2.0_User>

# 10. 开发预留字符

如开发过程中某些字符串字段需要使用符号进行分隔，然后解析的，请统一使用 “|” 符号。如所在地、地址或标签等等。即前端用户等输入时不能使用此字符。