|  |
| --- |
| 重庆紫光华山智安科技有限公司技术规范 |
| 异常处理及错误码规范 |
|  |
|  |
|  |

目 录

[1 简介 1-３](#_Toc524096457)

[1.1 异常处理规范 1-３](#_Toc524096458)

[1.2 本手册适用范 1-３](#_Toc524096459)

[2 Restful接口异常返回格式 2-３](#_Toc524096460)

[2.1 接口异常返回信息 2-３](#_Toc524096461)

[2.2 HTTP状态码 2-３](#_Toc524096462)

[3 异常code定义规范 3-４](#_Toc524096463)

[3.1 错误码定义规范 3-４](#_Toc524096464)

[3.2 通用错误编号定义 3-４](#_Toc524096465)

[4 Java异常处理规范 4-５](#_Toc524096466)

[4.1 编码规范 4-５](#_Toc524096467)

[4.1.1 运行时异常的捕捉 4-５](#_Toc524096468)

[4.1.2 try-cache作用域 4-６](#_Toc524096469)

[4.1.3 异常的捕获和异常的处理 4-６](#_Toc524096470)

[4.1.4 异常和事物 4-７](#_Toc524096471)

[4.1.5 Finally 4-７](#_Toc524096472)

[4.1.6 异常需要精准 4-７](#_Toc524096473)

# 简介

## 异常处理规范

异常处理规范定义了Restful接口返回异常的具体格式，以及错误码的定义规则。

## 本手册适用范围

本手册适用于所有研发人员。

# Restful接口异常返回格式

对于错误的响应，首先应遵循 HTTP 中关于 status code 的规范，并且返回规定的结构化数据。

## 接口异常返回信息

{

“error\_code”:” 000001”,

“message”:”接口停用”

}

字段含义解释

error\_code：错误码

message：异常消息

## HTTP状态码

为每个请求返回适当的状态码 参见 status code

200 OK，当GET请求成功完成，DELETE或者PATCH请求同步完成。

201 Created，对于那些要服务器创建对象的请求来说，资源已创建完毕。

202 Accepted，POST，DELETE或者PATCH请求提交成功，稍后将异步的进行处理。

204 No Content，Response中包含一些Header和一个状态行， 但不包括实体的主题内容（没有response body）

304 Not Modified，客户的缓存资源是最新的， 要客户端使用缓存

400 Bad Request

401 Unauthorized: 请求失败，因为用户没有进行认证

403 Forbidden: 请求失败，因为用户被认定没有访问特定资源的权限

405 Method Not Allowed：不支持该 Request 的方法

409 Conflict：发出的请求在资源上造成了一些冲突

429 Too Many Requests: 请求频率超配，稍后再试。

500 Internal Server Error: 服务器遇到一个错误，使其无法为请求提供服务

501 Not Implemented：客户端发起的请求超出服务器的能力范围(比如，使用了服务器不支持的请求方法)时，使用此状态码。

502 Bad Gateway：代理使用的服务器遇到了上游的无效响应

503 Service Unavailable：服务器目前无法为请求提供服务，但过一段时间就可以恢复服务

# 异常code定义规范

## 错误码定义规范

错误信息使用错误码（error\_code） + 错误信息（message）的方式。定义的错误码能让使用者清晰的知道，发生错误的系统、错误的类型以及错误的重要级别。因此错误码分为两段系统编号 + 错误编号，如 01000001。

系统编号：前两位01表示系统编号，如01表示VMS，02表示IMS。

错误编号：

编号长度为六位，第一位为错误码类型（0通用异常（统一定义）、1自定义异常），第二位为严重级别（0未定义严重级别、1普通提示信息、2错误、3致命错误， 其它类型暂时保留不用），后四位为系统错误码编码（建议:可以采用前两位为模块编号，后两位为模块异常编号）。

## 通用错误编号定义

000001 系统内部异常 000002 被禁止的IP

000003 当前IP请求超过限制 000004 错误的IP地址

000005 IP地址不能为空 000006 未知的请求源

000007 请求超过次数限制 000008 接口停用

000009 接口维护 000010 版本号错误

000011 域名错 000012 网络错误

000013 未知错误 000014 内部错误

000015 请求超时 000016 内存溢出

000017 数据源异常 000018 用户不存在

000019 用户没有权限 000020 当前操作频繁

000021 无数据 000022 资源已经存在

000023 查询无结果 000024 ID不能为空

000025 数据更新中 000026 参数校验错误

000027 手机号码错误 000028 身份证错误

000029 邮箱错误 000030 经纬度错误

000031 请求路径错误 000032 数据格式错误

000033 时间参数错误 000034 车牌号错误

000035 摄像头编号错误 000036 回调地址URL错误

000037 时间格式必须是ISO-8601 000038 页码超出限制

000039 分页大小超出限制 000040 接口不存在

000041 解密失败 000042 缺少必要参数

000043 下载失败 000044 未识别文件

000045 文件大小超过限制 000046 文件上传失败

000047 文件地址不存在 000048 账号密码错误

# Java异常处理规范

## 编码规范

### 运行时异常的捕捉

不要捕获Java类库中定义的继承自 RuntimeException 的运行时异常类，如：IndexOutOfBoundsException / NullPointerException ，这类异常由程序员预检查来规避，保证程序健壮性。

反例:

try {

obj.method()

} catch(NullPointerException e){

//TODO

}

正例:

if(null != obj) {

//TODO

}

### try-cache作用域

不要对大段代码进行try-catch，catch 时请分清稳定代码和非稳定代码，稳定代码指的是无论如何不会出错的代码。对于非稳定代码的 catch 尽可能进行区分异常类型，再做对应的异常处理。

正例:

map.put("status", 500);

try{

//代码省略

map.put("message", "success！");

map.put("status", 200);

} catch (UnknownHostException e) {

//域名错误

map.put("message", "网络异常");

LOG.error("网络异常", url);

} catch (SocketTimeoutException e) {

//超时

map.put("message", "请求地址超时");

LOG.error("请求地址超时", url);

}

return map;

反例：

try {

//此处省略

} catch(Exception e){

//TODO

}

### 异常的捕获和异常的处理

捕获异常是为了处理它，不要捕获了却什么都不处理而抛弃之，如果不想处理它，请将该异常抛给它的调用者。最外层的业务使用者，必须处理异常，将其转化为用户可以理解的内容。

正例：

try{

//代码省略

} catch (UnknownHostException e) {

//域名错误

map.put("message", "请输入正确的网址");

LOG.error("网络地址异常", url);

} catch (SocketTimeoutException e) {

//超时

map.put("message", "请求地址超时");

LOG.error("请求地址超时", url);

}

反例:

try{

//代码省略

} catch (Exception e) {

System.out.println("代码出现异常");

}

### 异常和事物

有 try 块放到了事务代码中，catch 异常后，如果需要回滚事务，一定要注意手动回滚事务。在有事务的代码中，catch 异常后，做描述异常等处理后，如果需要事务生效，请把异常继续抛出

### Finally

如果有对IO 流和资源做操作，必须逐一关闭IO 流和资源对象（从里层到外层），有异常也要做处理。不能在 finally 块中使用 return ，finally 块中的 return 返回后方法结束执行，不会再执行 try 块中的 return 语句，对于IO，流等closable的对象，推荐使用 try with resources的方式。

### 异常需要精准

捕获异常与抛异常，必须是完全匹配，或者捕获异常是抛异常的父类。