智慧型 BSS 业务规范

——web 层设计

2018 年 09 月

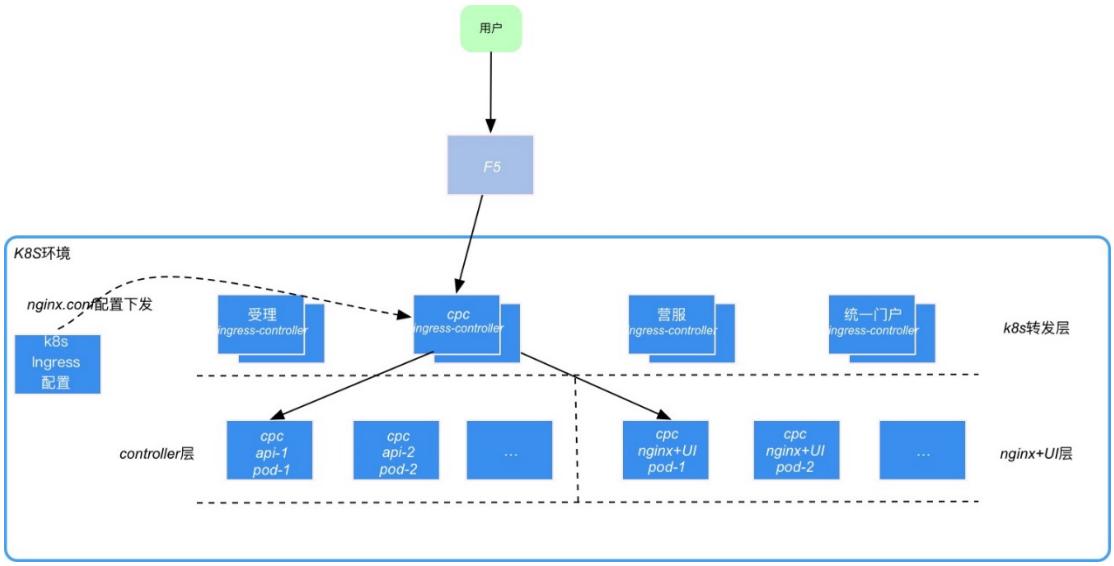
更改履历

版本号	更改时间	更改简要描述	更改人	批准人
V1.0	2018-9-21	初稿		

注：更改人除形成初稿，以后每次修改在未批准确认前均需采用修订的方式进行修改。

目录

1 架构图


用户
F5
K8S环境
nginx.conf配置下发
k8s
Ingress
配置
受理
ingress-controller
cpc
ingress-controller
营服
ingress-controller
统一门户
ingress-controller
k8s转发层
controller层
cpc
api-1
pod-1
cpc
api-2
pod-2
...
cpc
nginx+UI
pod-1
cpc
nginx+UI
pod-2
...
nginx+UI层

2 认证设计

采用 spring security+spring session+ctg cache 架构

2.1 通过 HeaderHttpSessionStrategy 生成 x-auth-token，客户端每次请求都需携带 x-auth-token，如没带会报 403，Access Denied

2.2 采用 GenericJackson2JsonRedisSerializer 序列化 SecurityContextImpl 对象及自定义 session 对象

2.3 session attribute 设计：

All:map--所有系统公用，包括：工号详情，员工详情

ORDER:map--受理中心专用，包括：客户

CPC:map--CPC 专用

2.4 用户退出，清空 session（"DEL""spring:session:sessions:expires:token"），对应的 spring:session:sessions:token 大对象没有删除，等待超时自动清空

2.5 验证 token 是否有效：

方法一：spring session 集成 ctg cache 由框架来验证

方法二：调用集团 ctg cache api 查询（"get""spring:session:sessions:expires:token"）

2.6 http head 带上 x-sysUserCode:工号，校验传进来的工号跟 session 中的工号是否一致，不一致报错

2.7 禁止随意调用 session.invalidate()，此方法会做 session 清空，禁止抛出 AuthenticationException 相关异常，此方法 security 框架会捕捉此异常并做 session 清空

3 代码规范

3.1 url 格式：/系统标示/系统模块/资源表达式，例如：

/bss-order/api-order-query/customerOrder/123455

备注：

/api-*:表示 ajax 请求后台服务，/order/order-query:表示静态资源

/api-order-query:ngnix 路由到 order-query-controller 后台服务，controller 层的 path 只需描述资源表达式

/资源表达式：参考集团 OpenApi

3.2 controller 统一返回 json 格式，包含正常/异常情况，见下面 restful 格式

3.3 充分利用 OpenApi 中的 api

3.4 smt-bss-centralized-authenticate 模块提供统一的认证，接口返回，session 控制，异常处理，json 过滤，引用如下配置：

```
@Import({WebSecurityConfig.class,
 HttpSessionConfig.class,
 MyUserDetailsService.class,
 MyResponseBodyAdvice.class,
 RestExceptionHandler.class,
 MyBasicErrorController.class,
 MyFilterConfig.class,
 MyJacksonConfig.class})
```

4 登录认证

4.1 支持用户名密码+验证码+读卡+人脸识别，目前只支持验证码方式

4.2 验证过程

4.2.1 校验用户：此过程主要是对用户的用户名和密码进行合法性校验，password 需要经过 MD5 加密处理。用户校验请求不经过 security 验证，controller 层调用基础中心服务获取工号基本信息（服务编码：00.1077.qrySystemUserBySysUserCode），校验密码。

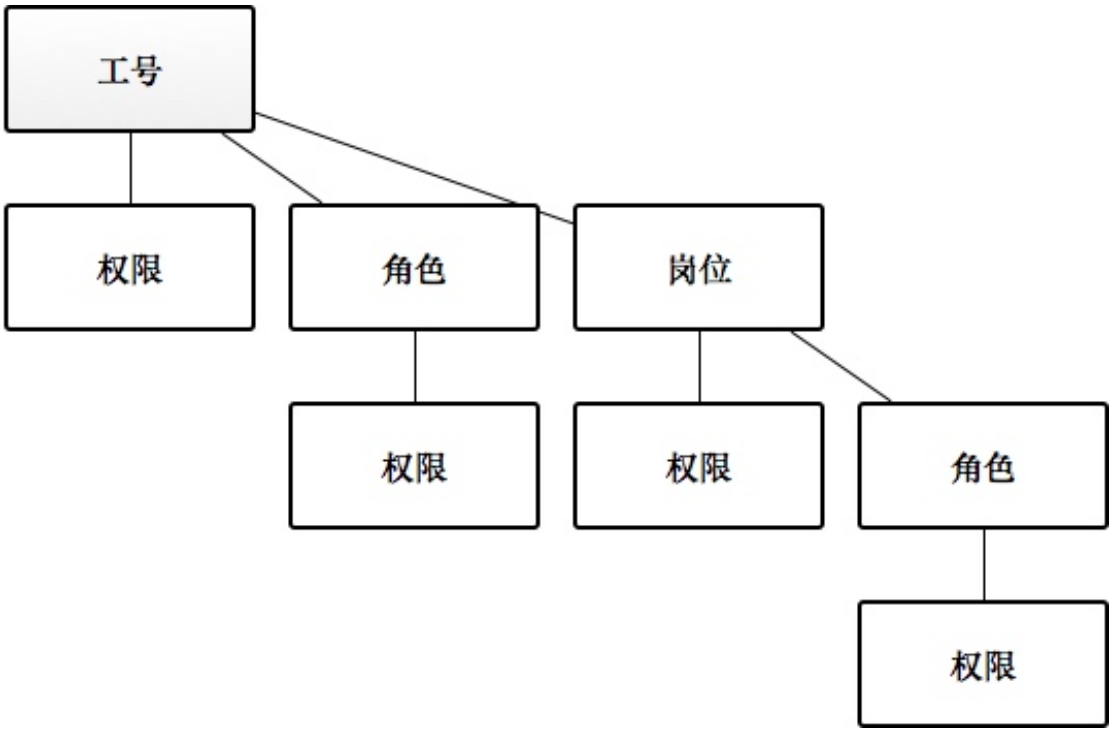
4.2.2 获取验证码：用户校验成功后跳转到手机获取验证码页面，手机号是系统用户绑定的手机号。获取验证码请求不经过 security 验证，controller 层调用基础运营中心发送验证码服务（服务编码 00.1077.sendCheckMessage），该服务会向系统用户绑定的手机号发送验证码，发送成功将验证码写入缓存中，并设置超时时间（默认 5 分钟）

4.2.3 登录认证：获取验证码成功后，客户端传入用户名，密码，验证码，通过 Spring Security 统一认证，验证成功，将工号详情，员工详情等写入 session 中，查询配置中心将版本号写入 cookie 中

4.2.4 跳转门户工作台：认证成功，通过 session 中的工号详情，获取相应的权限（功能菜单，功能组件）

4.2.5 退出登录：清空缓存

5 权限控制



5.1 工号详情接口返回的权限列表只包含授权允许的，由客户端通过变量的方式控制

5.2 工号->权限 大于 工号->角色->权限 大于 工号->岗位->权限 大于 工号->岗位->角色->权限

5.3 权限控制功能菜单，功能组件逻辑：判断是否存在相同的 URL，如存在，判断对应的权限是否允许，相同权限之间做优先级判断，不同权限间直接过滤

例如：权限控制功能菜单逻辑：

A 权限：1，2，3，4，5 --允许

B 权限：3 --禁止

最终返回功能菜单：1，2，4，5

5.4 后台 filter 会统一拦截所有的 URL，判断是否有权限访问

所有的 URL 都需要在基础中心配置，配置规范：

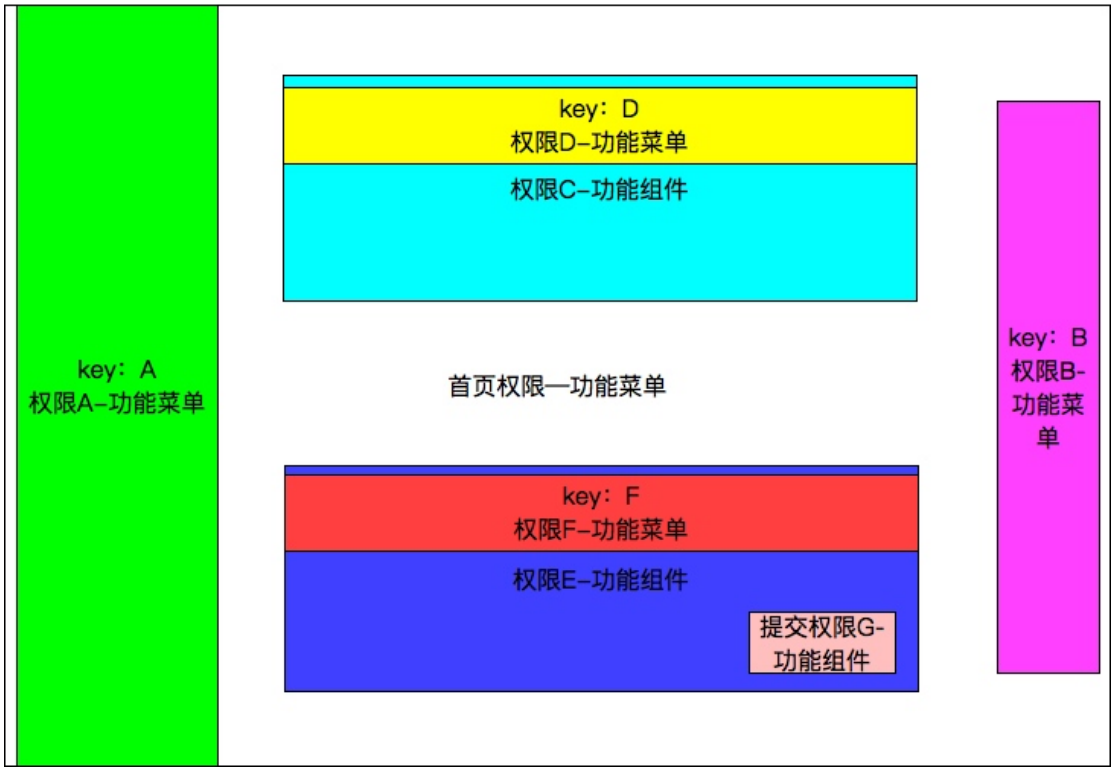
权限名称	权限描述	归属系统	权限类型	权限管控类别	关联功能类型	菜单/组件名称	上级菜单/组件	菜单/组件类型	菜单级别	菜单排序	菜单/组件 URL 地址	菜单/组件描述
受理中心_首页	受理中心首页界面	受理中心	功能权限	普通权限	功能菜单	首页		目录菜单	0	0	/order/home	
受理中心_首	受理中心首	受理中心	功能权限	普通权限	功能组件	轨迹账单推荐	首页	模块组件	1	1	xxx	

权限名称	权限描述	归属系统	权限类型	权限管控类别	关联功能类型	菜单/组件名称	上级菜单/组件	菜单/组件类型	菜单级别	菜单排序	菜单/组件 URL 地址	菜单/组件描述
页	界面					模块						
受理中心-首页	受理中心首页界面	受理中心	功能权限	普通权限	功能菜单	轨迹	轨迹账单推荐模块	叶子菜单	2	1	xxx	
受理中心-首页	受理中心首页界面	受理中心	功能权限	普通权限	功能菜单	账单	轨迹账单推荐模块	叶子菜单	2	2	xxx	
受理中心	受理中心	受理中心	功能权限	普通权限	功能菜单	推荐	轨迹账单	叶子菜单	2	3	xxx	

权限名称	权限描述	归属系统	权限类型	权限管控类别	关联功能类型	菜单/组件名称	上级菜单/组件	菜单/组件类型	菜单级别	菜单排序	菜单/组件 URL 地址	菜单/组件描述
-首页	首页界面						推荐模块					
受理中心-首页-账单-特殊权限	受理中心首页账单界面，指定岗位或角色	受理中心	功能权限	普通权限	功能菜单	账单	轨迹账单推荐模块	叶子菜单	2	3	xxx	

权限名称	权限描述	归属系统	权限类型	权限管控类别	关联功能类型	菜单/组件名称	上级菜单/组件	菜单/组件类型	菜单级别	菜单排序	菜单/组件 URL 地址	菜单/组件描述
	禁止看到											

6 前端跟权限如何结合



6.1 功能菜单：前端通过后台返回的 key 做循环遍历

6.2 功能组件：前端默认先不做控制，全部显示，当有需要，在单独配置权限，前端做相应的修改

7 Session Object 存储对象

目前 session 中的对象都是一些基础对象，后续可以按需添加

key	类型
ALL	Map

7.1 工号详情

key	类型
systemUserDetail	SystemUserDetail

SystemUserDetail 对象：

属性名称	类型	描述
sysUserId	Long	系统用户标识
staffId	Long	员工标识
sysUserCode	String	账号
password	String	密码
pwdErrCnt	Long	密码错误次数限制
pwdSmsTel	Long	验证码短信通知手机号

属性名称	类型	描述
pwdStatus	String	密码状态
pwdNewtime	DateTime	新密码生成时间
pwdEffectDays	Long	密码有效天数
regionId	Long	区域标识
systemInfoId	Long	归属系统
limitCount	Long	登录次数限制
loginedNum	Long	当前登录次数
sysUserDesc	String	系统用户描述
effDate	DateTime	生效时间
expDate	DateTime	失效时间
statusCd	String	状态
statusDate	DateTime	状态时间
createDate	DateTime	创建时间
createStaff	Long	创建人
updateDate	DateTime	修改时间
updateStaff	Long	修改人
actType	String	通用数据操作类型
privilegeDetails	List	权限详情
systemPostDetails	List	系统岗位详情
systemUserRoleDetails	List	系统用户角色详情

systemPostDetails 对象：

属性名称	类型	描述
sysPostId	Long	系统岗位标识
sysPostCode	String	系统岗位编码
sysPostName	String	系统岗位名称
sysPostType	String	系统岗位类型
sysPostDesc	String	系统岗位描述
initFlag	Long	是否系统初始数据
orgId	Long	系统岗位管理组织标识
regionId	Long	区域标识
systemInfoId	Long	归属系统
statusCd	String	系统岗位状态
statusDate	DateTime	状态时间
createDate	DateTime	创建时间
createStaff	Long	创建人
updateDate	DateTime	修改时间
updateStaff	Long	修改人
actType	String	通用数据操作类型
privilegeDetails	List	权限详情
systemRolesDetails	List	系统角色详情

systemUserRoleDetail 对象：

属性名称	类型	描述

属性名称	类型	描述
sysUserRoleId	Long	系统用户角色标识
sysRoleId	Long	系统角色标识
sysUserId	Long	系统用户标识
effDate	DateTime	生效时间
expDate	DateTime	失效时间
statusCd	String	状态
statusDate	DateTime	状态时间
createDate	DateTime	创建时间
createStaff	Long	创建人
updateDate	DateTime	修改时间
updateStaff	Long	修改人
actType	String	通用数据操作类型
systemRolesDetail	SystemRolesDetail	系统角色详情

systemRolesDetail 对象：

属性名称	类型	描述
sysRoleId	Long	系统角色标识
sysRoleName	String	系统角色名称
sysRoleCode	String	系统角色编码
sysRoleType	String	系统角色类型
sysRoleDesc	String	系统角色描述

属性名称	类型	描述
initFlag	Long	是否系统初始数据
regionId	Long	区域标识
systemInfold	Long	归属系统
statusCd	String	状态
statusDate	DateTime	状态时间
createDate	DateTime	创建时间
createStaff	Long	创建人
updateDate	DateTime	修改时间
updateStaff	Long	修改人
actType	String	通用数据操作类型
privilegeDetails	List	权限详情

privilegeDetail 对象：

属性名称	类型	描述
privId	Long	权限标识
privCode	String	权限编码
privName	String	权限名称
privType	String	权限类型
privDesc	String	权限描述
privManageClass	String	权限管控类别
effDate	DateTime	生效时间

属性名称	类型	描述
expDate	DateTime	失效时间
systemInfoId	Long	归属系统
statusCd	String	状态
statusDate	DateTime	状态时间
createDate	DateTime	创建时间
createStaff	Long	创建人
updateDate	DateTime	修改时间
updateStaff	Long	修改人
actType	String	通用数据操作类型
privFuncRelDetails	List	权限包含功能详情

privFuncRelDetails 对象：

属性名称	类型	描述
privFuncRelId	Long	权限包含功能标识
privId	Long	权限标识
privRefType	String	关联功能类型
privRefId	String	关联功能标识
effDate	DateTime	生效时间
expDate	DateTime	失效时间
systemInfoId	Long	归属系统
statusCd	String	状态

属性名称	类型	描述
statusDate	DateTime	状态时间
createDate	DateTime	创建时间
createStaff	Long	创建人
updateDate	DateTime	修改时间
updateStaff	Long	修改人
actType	String	通用数据操作类型
funcMenu	FuncMenu	功能菜单

funcMenu 对象：

属性名称	类型	描述
menuId	Long	菜单标识
menuName	String	菜单名称
menuType	String	菜单类型
menuLevel	Long	菜单级别
menuIndex	Long	菜单排序
parMenuId	Long	上级菜单标识
menuDesc	String	菜单描述
urlAddr	String	菜单 URL 地址
regionId	Long	区域标识
systemInfoId	Long	归属系统
statusCd	String	状态

属性名称	类型	描述
statusDate	DateTime	状态时间
createDate	DateTime	创建时间
createStaff	Long	创建人
updateDate	DateTime	修改时间
updateStaff	Long	修改人
actType	String	通用数据操作类型

7.2 员工详情

key	类型
staffDetail	StaffDetail

案例：


```
{
  "resultCode" : "0000",
  "resultObject" : {
    "staffDetail" : {
      "staffAttrs" : [
        {
          "staffId" : 289,
          "attrValue" : "11",
          "staffAttrId" : 290,
          "attrId" : 50000016,
          "statusCd" : "1000"
        },
        {
```

```
      "staffId" : 289,
      "attrValue" : "30",
      "staffAttrId" : 291,
      "attrId" : 50000018,
      "statusCd" : "1000"
    },
    {
      "staffId" : 289,
      "attrValue" : "10",
      "staffAttrId" : 292,
      "attrId" : 50030003,
      "statusCd" : "1000"
    }
  ],
  "staffContactInfos" : [],
  "staffDesc" : "",
  "createStaff" : -1,
  "createDate" : 1528338770000,
  "salesstaffCode" : "Y33050024036",
  "statusDate" : 1528338770000,
  "staffAccount" : "72QW873",
  "commonRegionId" : 8330500,
  "statusCd" : "1000",
  "orgId" : 204697889,
  "staffCode" : "Y33050024036",
  "staffId" : 289,
  "partyId" : -1,
  "staffName" : "金珍云",
  "staffOrgRels" : [
    {
      "relType" : "30",
```

```
          "staffId" : 289,
          "statusCd" : "1000",
          "staffOrgRelId" : 293,
          "orgId" : 204654911
        }
      ],
      "organization" : {
        "createStaff" : 11,
        "createDate" : 1311884146000,
        "salesorgCode" : "1-7TLT",
        "updateStaff" : 11,
        "statusDate" : 1370686000000,
        "regionId" : 8330500,
        "statusCd" : "1000",
        "orgType" : "1000",
        "orgId" : 204697889,
        "updateDate" : 1311884146000,
        "orgCode" : "1-7TLT",
        "partyId" : -1,
        "orgName" : "湖州电信",
        "orgSubtype" : "1200",
        "villageFlag" : "1100"
      }
    }
  }
}
```


7.3 允许访问的 url 列表

key	类型

key	类型
urlPrivilegeSet	List<String>

8 API 描述

8.1 校验用户

8.1.1 功能描述

通过用户名密码校验用户的合法性

8.1.2 请求说明

请求方式：POST

请求 URL：/login/check

8.1.3 请求参数

字段	字段类型	字段说明
username	string	用户名
password	string	密码，MD5 加密

8.1.4 返回结果

例如：

```
{
```

```
    "code": "0000",
    "message": "用户校验成功",
    "timestamp": 1537166287419,
    "status": 200,
    "path": "/login/check"
}
```

8.2 发送验证码

8.2.1 功能描述

通过用户名密码校验用户，并成功发送验证码

8.2.2 请求说明

请求方式：POST

请求 URL：/login/random-code

8.2.3 请求参数

字段	字段类型	字段说明
username	string	用户名
password	string	密码，MD5 加密

8.2.4 返回结果

例如：

```
{
```

```
    "code": "0000",
    "message": "模拟验证码",
    "timestamp": 1537166995211,
    "status": 200,
    "path": "/login/random-code"
}
```


8.3 校验用户

8.3.1 功能描述

通过用户名密码验证码登录认证

8.3.2 请求说明

请求方式：POST

请求 URL：/login

8.3.3 请求参数

字段	字段类型	字段说明
username	string	用户名
password	string	密码，MD5 加密
code	string	验证码

8.3.4 返回结果

例如：

```
{
    "code": "0000",
    "message": "认证成功",
    "timestamp": 1537166999591,
    "status": 200,
    "path": "/login"
}
```

8.4 门户首页

8.4.1 功能描述

登录成功，跳转至门户首页

8.4.2 请求说明

请求方式：GET

请求 URL：/index

head：x-auth-token

8.4.3 返回结果

data 对象：

属性名称	类型	描述
sysUserId	Long	系统用户标识
sysUserCode	String	系统用户名称

属性名称	类型	描述
staffName	String	工号对应的员工姓名
userImg	String	系统用户头像
workBenchVos	List	工作台

workBenchVos 对象：

属性名称	类型	描述
img	String	工作台图标
title	String	工作台名称
explainItems	List	工作台包含子集（菜单）

explainItems 对象：

属性名称	类型	描述
menuName	String	菜单名称
urlAddr	String	菜单跳转地址

例如：


```
{
"code": "0000",
"data": {
    "sysUserId": 3950,
    "staffName": "陈小娇",
    "userImg": "",
    "sysUserCode": "shegw",
```

```
    "workBenchVos": [
        {
            "img": "",
            "title": "IT 支撑工作台",
            "explainItems": [
                {
                    "menuName": "CPC 配置中心",
                    "urlAddr": "/bss-cpc/#/home"
                }
            ]
        }
    ]
},
"message": "成功",
"timestamp": 1537258606716,
"status": 200,
"path": "/index"
}
```


8.5 退出登录

8.5.1 功能描述

安全退出登录

8.5.2 请求说明

请求方式：GET

请求 URL：/login/logout

head：x-auth-token

8.5.3 返回结果

例如：

```
{
    "code": "0000",
    "message": "退出成功",
    "timestamp": 1537167768915,
    "status": 200,
    "path": "/login/logout"
}
```

9 常量值描述

9.1 restful 统一返回 json 格式

字段	字段类型	字段说明
code	string	成功失败 code
data	object	返回对象
message	string	描述（包含成功，失败信息）

字段	字段类型	字段说明
exception	string	异常类名
timestamp	long	请求时间
status	int	http 状态码
path	string	请求路径

9.2 code 值

值	描述
0000	成功
0001	服务异常
0002	服务逻辑异常
0003	dubbo 调用异常
0004	登陆超时
0005	没有权限访问
0006	未登录
0007	未匹配 URL

9.3 status 值

值	描述	方法
200	OK–这个是标准的成功返回	ALL
201	Created–请求已经被实现，而且有一个新的资源已经依据请求的需要而建立，且其 URI 已经随 Location 头信	POST,PUT

值	描述	方法
	息返回。	
202	Accepted–请求被成功接受并且会异步的处理	POST,PUT,DELETE,PATCH
204	No Content–服务器成功处理了请求，但不需要返回任何实体内容，并且希望返回更新了的元信息。	POST,DELETE, PATCH
207	Multi–Status–由 WebDAV(RFC 2518)扩展的状态码，代表之后的消息体将是一个 XML 消息，并且可能依照之前子请求数量的不同，包含一系列独立的响应代码。参见Must:在批量请求中使用207	POST
301	Moved Permanently–被请求的资源已永久移动到新位置，并且将来任何对此资源的引用都应该使用本响应返回的若干个 URI 之一。	All
303	See Other – 对应当前请求的响应可以在另一个 URI 上被找到，而且客户端应当采用用 GET 的方式访问那个资源。	PATCH, POST, PUT, DELETE
304	Not Modified – 如果客户端发送了一个带条件的 GET 请求且该请求已被允许，而文档的内容(自上次访问以来或者根请求的条件)并没有改变，则服务器应当返回这个状态码	GET
400	Bad request – 语义有误，当前请求	ALL

值	描述	方法
	无法被服务器理解。除非进行修改，否则客户端不应该重复提交这个请求。请求参数有误	
401	Unauthorized – 当前请求需要用户验证	ALL
403	Forbidden – 用户没有被授权访问该资源	ALL
404	Not found – 请求失败，请求所希望得到的资源未被在服务器上发现	ALL
405	Method Not Allowed – 请求行中指定的请求方法不能被用于请求相应的资源。该响应必须返回一个 Allow 头信息用以表示出当前资源能够接受的请求方法的列表	ALL
406	Not Acceptable – 请求的资源的内容特性无法满足请求头中的条件，因而无法生成响应实体	ALL
408	Request timeout – 请求超时。客户端没有在服务器预备等待的时间内完成一个请求的发送。客户端可以随时再次提交这一请求而无需进行任何更改	ALL
409	Conflict – 由于和被请求的资源的当前状态之间存在冲突，请求无法完成。这个代码只允许用在这样的情况下才能被使用:用户被认为能够解决冲	POST , PUT, DELETE, PATCH

值	描述	方法
	突，并且会重新提交新的请求。该响应应当包含足够的信息以便用户发现冲突的源头	
410	Gone – 被请求的资源在服务器上已经不再可用，而且没有任何已知的转发地址。例如当访问一个已经被要求删除的资源	ALL
412	Precondition Failed – 服务器在验证在请求的头字段中给出先决条件时，没能满足其中的一个或多个。这个状态码允许客户端在获取资源时在请求的元信息(请求头字段数据)中设置先决条件，以此避免该请求方法被应用到其希望的内容以外的资源上	PUT, DELETE, PATCH
415	Unsupported Media Type – 对于当前请求的方法和所请求的资源，请求中提交的实体并不是服务器中所支持的格式，因此请求被拒绝	POST , PUT, DELETE, PATCH
423	Locked – 当前资源被锁定	PUT, DELETE, PATCH
428	Precondition Required – server requires the request to be conditional (e. g. to make sure that the “lost update problem” is avoided)	ALL
429	Too many requests – 客户端没有处理速率限制并且发送了太多的请求.	ALL

值	描述	方法
	See Must: Use 429 with Headers for Rate Limits	
500	Internal Server Error – 服务器遇到了一个未曾预料的状况，导致了它无法完成对请求的处理。一般来说，这个问题都会在服务器端的源代码出现错误时出现	ALL
501	Not Implemented – 服务器不支持当前请求所需要的某个功能。当服务器无法识别请求的方法，并且无法支持其对任何资源的请求。（通常用于暗示可用的新功能，例如新的特性）	ALL
503	Service Unavailable – 由于临时的服务器维护或者过载，服务器当前无法处理请求。这个状况是临时的，并且将在一段时间以后恢复。如果能够预计延迟时间，那么响应中可以包含一个 Retry-After 头用以标明这个延迟时间。如果没有给出这个 Retry-After 信息，那么客户端应当以处理 500 响应的方式处理它	ALL