ALIBABA CLOUD

Alibaba Cloud

API 网关

常见问题

文档版本：20220513

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.API网关 - 事件总线服务关联角色 05
2.如何获取错误信息 06
3.错误代码表 07
4.Invalid Signature 23
5.Invalid Url 24
6.Unauthorized 25
7.ApiGateway服务关联角色 26
8.API网关 - 监控服务关联角色 27
9.后端服务调不通 29
10.API定义 33
10.1. API网关是否支持gzip数据压缩，是否需要添加请求头Accept-Enc... 33
11.调用API 34
11.1. 请求中编码问题 34

1.API网关 - 事件总线服务关联角色

本文为您介绍API网关 - 事件总线服务关联角色（AliyunServiceRoleForApigatewayPutEventsToEventBridge）的应用场景以及如何删除API网关 - 事件总线服务关联角色。

背景信息

API网关 - 事件总线服务关联角色（AliyunServiceRoleForApigatewayPutEventsToEventBridge）是API网关为了访问用户的事件总线产品，上传事件到用户的事件总线而提供的RAM角色。更多关于服务关联角色的信息请参见服务关联角色。

应用场景

用户在创建事件总线类型的后端服务时，API网关会提示用户创建事件总线服务关联角色，在后续操作中，用户在API网关创建后端为事件总线的API并调用，API网关在上传事件到用户的事件总线时需要使用服务关联角色（AliyunServiceRoleForApigatewayPutEventsToEventBridge）获取访问权限。

AliyunServiceRoleForApigatewayPutEventsToEventBridge 介绍

角色名称：AliyunServiceRoleForApigatewayPutEventsToEventBridge 角色权限策略：AliyunServiceRolePolicyForApigatewayPutEventsToEventBridge 权限说明：

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "eventbridge:PutEvents",
      "Resource": "*"
    }
  ]
}
```

删除服务关联角色

如果您需要删除AliyunServiceRoleForApigatewayPutEventsToEventBridge（服务关联角色），需要先删除依赖这个服务关联角色的后端服务和API。操作步骤如下：

1. 进入API网关控制台，在左侧导航栏选择**开放API-后端服务**；
2. 在搜索框，通过后端服务类型**EventBridge**搜索后端服务，在右侧操作列点击**删除**；
3. 进入RAM控制台，点击左侧菜单 `RAM 角色管理` ，找到角色 `AliyunServiceRoleForApigatewayPutEventsToEventBridge` 删除。

2.如何获取错误信息

所有的 API 请求只要到达了网关，网关就会返回请求结果信息。

用户需要查看返回结果的头部，即 Header 部分。其中 X-Ca开头的均为网关返回，比较重要信息是：

```
// 请求唯一ID，请求一旦进入API网关应用后，API网关就会生成请求ID并通过响应头返回给客户端，建议客户端与后端服务都记录此请求ID，可用于问题排查与跟踪
X-Ca-Request-Id: 7AD052CB-EE8B-4DFD-BBAF-EFB340E0A5AF
// API网关返回的错误消息，当请求出现错误时API网关会通过响应头将错误消息返回给客户端
X-Ca-Error-Message: Invalid Parameter Required `field1`
// API网关系统错误码，当请求出现错误被网关拦截后，由API网关提供的错误码：（经典网络实例无此头）
X-Ca-Error-Code: I400MP
```

在应答的 `Header` 中获得 `X-Ca-Error-Code` 与 `X-Ca-Error-Message` 可以基本明确报错原因，而 `X-Ca-Request-Id` 可以用于在日志服务中查询请求日志、通过控制台查询结果、或提供给支持人员进行日志排查。

`X-Ca-Error-Code` 可以查找错误代码表来获取更详细的拍错解释。

3.错误代码表

1. API网关错误码表（VPC实例）

本章节的错误代码表适用于 `VPC共享实例` 和 `VPC专享实例` ，如果您使用的是 `经典网络实例` 请参见下一节。

- 当客户端收到的应答中 `X-Ca-Error-Code` 头不为空，表示应答码由API网关产生，错误码由一个6位长度的字符描述，请参考下表，而 `X-Ca-Error-Message` 表示错误的应答信息，用于描述改场景下更详细的一些错误信息。
- 如果 `X-Ca-Error-Code` 头为空，则表示这个Http应答码由后端服务产生，API网关透传了来自后端的错误信息。

错误代码	Http状态码	Message	描述
I400HD	400	Invalid Header `${HeaderName}` ${Reason}	HTTP请求头非法
I400MH	400	Header `${HeaderName}` is Required	缺少HTTP请求头
I400BD	400	Invalid Body: ${Reason}	HTTP请求包体非法
I400PA	400	Invalid Request Path `${Reason}`	HTTP请求路径非法
I405UM	405	Unsupported Method `${Reason}`	不支持的HTTP请求方法
I400RU	400	Invalid Request Uri `${Reason}`	HTTP请求Url非法
I403PT	403	Invalid protocol ${Protocol} unsupported	使用了API配置中不支持的协议，请检查API配置
I413RL	413	Request body too Large	请求包体过长
I413UL	413	Request URL too Large	请求URL过长

I400CT	400	Invalid Content-Type: `${Reason}`	非法的`Content-Type`
I404DO	404	Invalid Domain `${DomainName}`	未知的请求域名
I410GG	410	Group's instance invalid	请求了非法的实例，分组可能已经不属于当前实例
I400SG	400	Invalid Stage	请求了未知的环境
I404NF	404	API not found ${Reason}	根据请求的`Path`,`Method`在当前的环境中未找到API
X400PM	400	Invalid plugin meta ${PluginName} ${Reason}	插件元数据非法，请工单联系客服人员
X500ED	500	Expired api definition	API元数据定义非法，请工单联系客服人员
X500AM	500	Invalid Api Meta, try deploy again or contact us via ticket	API元数据定义非法，请工单联系客服人员
X403DG	403	Bad Domain or Group: ${Reason}	分组数据非法,请工单联系客服人员
B451DO	451	Unavailable Domain for Legal Reasons	域名因法律法规问题被禁
B451GO	451	Unavailable Group for Legal Reasons	分组因法律法规问题被禁
B403OD	403	Provider Account Overdue	API提供方欠费
A400AC	400	Invalid AppCode ${Reason}	当使用AppCode模式授权时，未找到AppCode

A400IK	400	Invalid AppKey	当使用`Key/Secret`签名授权时，未找到AppKey
A403IS	403	Invalid Signature, Server StringToSign:`${StringToSign}`	签名不匹配，请参考API网关签名文档
A403EP	403	App authorization expired	授权已过期
A403PR	403	Plugin Authorization Needed	需要插件授权
A400MA	400	Need authorization, `X-Ca-Key` or `Authorization: APPCODE ...` is required	需要使用`Key/Secret`签名授权或`AppCode`授权
I400I5	400	Invalid Content-MD5 ${Reason}	不匹配的`Content-MD5`
I400NC	400	X-Ca-Nonce is required	当设置了`使用X-Ca-Nonce防重放`选项时，必须提供`X-Ca-Nonce`头
S403NU	403	Nonce Used	检测到请求重放，请求的`X-Ca-Nonce`头重复
S403TE	403	X-Ca-Timestamp is expired	`X-Ca-Timestamp`头中提供的时间戳已过期
I400MP	400	Parameter `${ParameterName}` is required	API中配置的必选参数未传值
I400IP	400	Invalid parameter `${ParameterName}` ${Reason}	API中配置的参数值非法
I400JR	400	JWT required	未找到JWT参数

S403JI	403	Claim `jti` is required when `preventJtiReplay:true`	当在`JWT授权插件`中配置了防重放功能时，请求未提供有效的`jti`
A403SV	403	Claim `jti` in JWT is used	当在`JWT授权插件`中配置了防重放功能时，请求提供的`jti`已被使用
I400JD	400	JWT Deserialize Failed: `${Token}`	请求中提供的`JWT`解析失败
A403JT	403	Invalid JWT: ${Reason}	请求中提供的`JWT`非法
A403JK	403	No matching JWK, `${kid}` not found	请求`JWT`中的`kid`没有匹配的`JWK`
A403JE	403	JWT is expired at `${Date}`	请求中提供的`JWT`已过期
I400JP	400	Invalid JWT plugin config: ${JWT}	`JWT授权`插件配置错误
A403OL	403	OAuth2 Login failed: ${Reason}	
A403OU	403	OAuth2 Get User Info failed: ${Reason}	
A401OT	401	Invalid OAuth2 Access Token	
A401OM	401	OAuth2 Access Token is required	
T429ID	429	Throttled by INNER DOMAIN Flow Control, ${Domain} is a test domain, only 1000 requests per day	当使用默认二级域名访问时，限制1000次/天，请绑定正式域名以解除这个限制

T429IN	429	Throttled by INSTANCE Flow Control	触发当前实例的流控限制
T429GR	429	Throttled by GROUP Flow Control	触发当前分组的流控限制
T429PA	429	Throttled by API Flow Control	触发插件上的默认API流控
T429PR	429	Throttled by PLUGIN Flow Control	触发插件的特殊流控
T429UP	429	Throttled by Usage Plan Flow Control	触发使用计划的流控
T429SR	429	Throttled by SERVER Flow Control	
T429MR	429	Too Many Requests, throttle by `${Description}`	
A403IP	403	Access denied by IP Control Policy	被`IP访问控制插件`阻止访问
A403IN	403	Access from internet is disabled ${Reason}	`API`或`API分组`禁止从公网访问
A403VN	403	Access from invalid VPC is disabled	来源VPC被阻止
A403AC	403	Access Control Forbidden by ${RuleName}	被`授权控制`插件阻止
A403CO	403	Cross origin resource forbidden ${Domain}	被CORS策略阻止访问
I404CO	404	Cross origin resource not found ${Method} - ${Path}	根据CORS预检请求中的Path与Method，无法找到API定义

I404CH	404	Content not cached, with `Cache-Control:only-if-cached`	
I404NR	404	${Resource} not found	
I404SR	404	Stage route missing: ${Reason}	
B403MO	403	Api Market Subscription overdue	API提供商欠费
B403MQ	403	Api Market Subscription quota exhausted	购买的云市场API配额已耗尽
B403ME	403	Api Market Subscription expired	API订购关系已过期
B403MI	403	Api Market Subscription invalid	API市场订购关系非法
D504RE	504	Backend domain `${Domain}` resolve failed	后端域名解析失败
D504IL	504	Backend domain `${Domain}` resolve to illegal address `${Address}`	后端域名解析结果非法
D504CO	504	Backend service connect failed `${Reason}`	后端连接失败，请检查安全组、后端服务器启动状态、或防火墙配置
D504CS	504	Backend http ssl connect failed `${Reason}`	后端HTTPS连接失败，请检查后端配置的协议与端口是否匹配
D504TO	504	Backend service request timeout	后端请求超时

X504VE	504	Backend service vpc mapped failed	后端VPC映射错误，请工单联系客服人员
D503BB	503	Backend circuit breaker busy	API被断路器阻止
D503CB	503	Backend circuit breaker open, ${Reason}	API处于熔断/断路器开状态，请检查后端性能
I508LD	508	Loop Detected	检测到环回调用
I404DD	404	Device id ${DeviceId} not found	当使用WebSocket双向通信调用时，DeviceId未找到
A403FC	403	Function Compute AssumeRole failed ${RequestId}:${Reason}	后端是函数计算时授权错误
D502FC	502	Function Compute response invalid: ${Reason}	后端是函数计算时，来自后端的应答非法
N502RE	502	Send Response IO Exception: ${Reason}	发送应答给客户端时报错，常见于客户端提前关闭连接或网络错误
X500ER	500	Service Internal Error	服务器内部错误，请工单联系工作人员
X503BZ	503	Service Busy	API网关服务忙，请稍后再试或工单联系工作人员
X504TO	504	Service timeout	API网关处理超时

部分错误代码可能随着升级或新功能的加入而改变。

2. API网关错误码表（经典网络实例）

2.1. 服务端错误码表

HttpCode是5xx，表示服务不可用。此时一般建议重试或联系商品页面的API服务商。

错误代码	Http 状态码	语义	解决方案
Internal Error	500	API 网关内部错误	建议重试。
Failed To Invoke Backend Service	500	底层服务错误	API 提供者底层服务错误，建议重试，如果重试多次仍然不可用，可联系 API 服务商解决。
Service Unavailable	503	服务不可用	建议稍后重试。
Async Service	504	后端服务超时	建议稍后重试。

2.2. 客户端错误码表

HttpCode为4xx，表示业务报错。此时一般为参数错误、签名错误、请求方式有误或被流控限制等业务类错误。建议详细查看错误码，针对性解决问题。

错误代码	Http 状态码	语义	解决方案
Throttled by USER Flow Control	403	因用户流控被限制	一般是由于 API 服务商设置的用户流控值导致被流控，可以联系 API 服务商协商放宽限制。
Api Prov	403	因用户流控被限制	一般是由于 API 服务商设置的用户流控值导致被流控，可以联系 API 服务商协商放宽限制。
Throttled by APP Flow Control	403	因APP流控被限制	一般是由于 API 服务商设置的 APP 流控值导致被流控，可以联系 API 服务商协商放宽限制。
Throttled by API Flow Control	403	因 API 流控被限制	一般是由于 API 服务商设置的 API 流控值导致被流控，可以联系 API 服务商协商放宽限制。

Throttled by DOMAIN Flow Control	403	因二级域名流控被限制，或因分组流控被限制	直接访问二级域名调用API，每天被访问次数上限1000次。每个分组被访问的QPS限制为500。
Quota Exhausted	403	调用次数已用完	购买的次数已用完。
Expiry of Authorization	403	已经超出api授权时间范围	重新授权，按需选择授权时间范围
Quota Expired	403	购买次数已过期	购买的次数已经过期。
User Arrears	403	用户已欠费	请尽快充值续费。
Empty Request Body	400	body 为空	请检查请求 Body 内容。
Invalid Request Body	400	body 无效	请检查请求 Body。
Invalid Param Location	400	参数位置错误	请求参数位置错误。
Unsupported Multipart	400	不支持上传	不支持上传文件。
Invalid Url	400	Url 无效	请求的 Method、Path 或者环境不对。
Invalid Domain	400	域名无效	请求域名无效，根据域名找不到 API。请联系 API 服务商。
Invalid HttpMethod	400	HttpMethod 无效	输入的 Method 不合法。
Invalid AppKey	400	AppKey 无效或不存在	请检查传入的 AppKey。注意左右空格的影响。
Invalid AppSecret	400	APP 的Secret 错误	检查传入的 AppSecret。注意左右空格的影响。
Timestamp Expired	400	时间戳过时	请核对请求系统时间是否为标准时间。

Invalid Timestamp	400	时间戳不合法	请参照请求签名说明文档
Empty Signature	401	签名为空	请传入签名字符串，请参照请求签名说明文档
Invalid Signature, Server StringToSign:%s	400	签名无效	签名无效。
Invalid Content-MD5	400	Content-MD5 值不合法	请求 Body 为空，但传入了 MD5 值，或 MD5 值计算错误。
Unauthorized	403	未被授权	APP 未获得要调用的 API 的授权。
Nonce Used	400	SignatureNonce 已被使用一次	SignatureNonce 不能被重复使用。
API Not Found	400	找不到 API	传入的 GroupId、Stage 等参数不正确，或已下线。
OpenID Connect Verify Fail: 243, Not found by keyId:%s	401	OpenID Connect 认证失败	检查下列项： 1.请检查您是否使用了您本地demo代码生成的token, token必须由授权API在部署环境生成。 2.获取授权API，业务API，这两者的安全认证都必须是OpenID Connect类型。 3.配置获取授权API时，KeyId和公钥一定要与后端代码实际使用的参数一致。

3. 管控OpenAPI错误代码表

当您调用API网关开放的管控OpenAPI时，如CreateAPI、ModifyAPI、DeleteAPI等，您可能遇到以下错误码。

3.1. 服务端错误码

HttpCode为5xx，表示服务不可用。此时建议重试。

错误代码	描述	Http 状态码	语义	解决方案
ServiceUnavailable	The request has failed due to a temporary failure of the server.	503	服务不可用	建议重试。
InternalError	The request processing has failed due to some unknown error, exception or failure.	500	内部错误	建议重试。

3.2. 客户端错误码

HttpCode为4xx，表示业务错误。一般为参数错误、权限限制、业务逻辑错误等问题，此时需要仔细查看错误码，并针对性解决问题。

错误代码	描述	Http 状态码	语义	解决
Repeated%s	The specified %s is repeated.	400	某参数重复（%s 是占位符，实际调用会给出明确的参数名或提示。）	建议按照提示修改重复的参数后重试。
RepeatedCommit	Resubmit request.	400	请求重复	请不要频繁提交请求。
Missing%s	The %s is mandatory for this action.	400	缺少参数 %s	根据具体返回补充缺少的参数，重试请求。
MissingAppIdOrAppOwner	AppId or AppOwner must have a valid value.	400	缺少参数 AppId 或者 AppOwner	参数 AppId 和 AppOwner 不能同时为空。
Invalid%s	The specified parameter %s value is not valid.	400	参数无效	根据返回提示的具体参数，查看相关参数约束，修改后重试。

NotFound%s	Cannot find resource according to your specified %s.	400	找不到资源	根据指定的参数%s找不到资源，建议检查%s是否正确。
InvalidFormat%s	The specified parameter %s value is not well formatted.	400	参数格式错误	建议根据实际返回提示，查看%s的格式要求，修改后重试。
Duplicate%s	The specified parameter %s value is duplicate.	400	参数重复	某请求参数不允许重复，建议检查修正后重试。
DependencyViolation%s	The specified %s has %s definitions.	400	参数依赖错误	指定参数被依赖，不能随意删除，请先解除依赖。
Forbidden%s	Not allowed to operate on the specified %s.	403	无权操作/操作禁止	您无权执行该操作。
NoPermission	User is not authorized to operate on the specified resource.	403	无权操作	RAM 鉴权不通过。
ExceedLimit%s	The specified %s count exceeds the limit.	400	超出个数限制	一般指用户账户下创建的 API、API 分组或者APP超出个数限制。
UserNotFound	The specified user can not be found.	404	指定用户找不到	根据输入的用户信息找不到该用户。
DomainCertificateNotFound	Cannot find the domain certificate.	400	指定域名证书不存在	建议检查传入的证书 ID 及名称。

DomainNotResolved	The specified domain has not been resolved.	400	指定域名未解析	需要将指定域名 CNAME 解析到分组的二级域名上，才能完成绑定。域名解析是在用户域名购买的网站上操作。
InvalidICPLicense	The specified domain have not got ICP license, or the ICP license does not belong to Aliyun.	400	域名备案不合格	要绑定的域名需要在阿里云做首次备案，已经在其他系统做备案的，需要在阿里云备案接入。备案接入需要获取备案号，在阿里云有 ECS 且具有公网 IP 的，每个 ECS 有5个备案号。
Invalid%s.LengthLimit	The parameter %s length exceeds the limit.	400	长度超限	参数%s超出长度限制，建议修正后重试。
InvalidApiDefault	The ApiDefault value exceeds limit.	400	流控 API 默认值超过限制值	该值数值不能超过1亿，与单位无关。如需上调请提工单申请。
InvalidAppDefault	The AppDefault value must smaller than the UserDefault and ApiDefault.	400	AppDefault 的值不符合规则	该值需要小于 API 流控值和用户流控值。
InvalidUserDefault	The UserDefault value must bigger than the AppDefault and smaller than the ApiDefault.	400	UserDefault 的值不符合规则	该值需要小于 API 流控值并大于 APP 流控值。
InvalidParamMapping	Parameters must be fully mapped.	400	参数映射无效	创建 API 时，前后端参数映射需要是全映射。即每个入参都需要配置后端参数名称。

InvalidOwnerAccount	OwnerAccount is invalid.	400	APP所有者账号无效	一般为操作授权时，输入目标用户的阿里云邮箱账号，该账号无效。建议检查修正后重试。
ServiceForbidden	Your Gateway service is forbidden by risk control.	400	API 网关服务被风控（应该使用户被风控吧）	请注意不要频繁请求，建议稍后重试。若重试后仍未解决，可提工单咨询。
ServiceUnOpen	Your Gateway service has not been opened.	400	服务未开通	建议到阿里云官网开通一下 API 网关服务。
ServiceInDept	Your API Gateway service is in dept.	400	（您的 API 网关）服务欠费	充值或者结算后重新开始使用。
EqualSignature	The new signature is the same as the old.	400	新密钥与旧的相等	修改后端签名密钥时，新设置的 Key 和 Secret 不能跟原来的一样。
CertificateNotMatch	The domain does not match the one in the certificate.	400	证书不匹配	指定域名与证书内域名不匹配。
CertificateKeyNotMatch	The certificate private key does not match the public key.	400	证书密钥不匹配	证书的公钥和私钥不匹配。
PrivateKeyEncrypted	The certificate private key is encrypted, please upload the unencrypted version.	400	密钥不能加密	证书私钥加密了，要求上传不加密的版本。
CertificateSecretKeyError	The certificate private key is invalid.	400	证书私钥错误	建议检查后重新传入。

InvalidApiServiceAddress	The specified service address is not valid.	400	API 后端服务地址无效	配置的 API 后端服务地址无效。

3.3. 客户端公用错误码

HttpCode为4xx，调用全阿里云产品的OpenAPI均可能遇到，表示业务错误。一般为请求格式错误、请求方式错误、必填参数丢失、参数格式错误、签名错误、流控限制等问题。此时需查看具体错误码，针对性解决问题。

报错场景	错误码	错误提示	状态码	建议
API 找不到。	InvalidApi.NotFound	Specified api is not found, please check your url and method.	404	请检查指定的 action 接口名称是否正确，注意大小写区分。
缺少必填参数。	Missing{ParameterName}	{ParameterName} is mandatory for this action.	400	指定参数为必填参数，请传入。
AccessKeyId 找不到	InvalidAccessKeyId.NotFound	Specified access key is not found.	404	请检查调用时是否传入正确的 AccessKeyId。
AccessKeyId 被禁用。	InvalidAccessKeyId.Inactive	Specified access key is disabled.	400	检查 AK 是否可用。
时间戳格式不对(Date 和 Timestamp)。	InvalidTimeStamp.Format	Specified time stamp or date value is not well formatted.	400	检查时间戳。
用户时间和服务器时间超过15分钟。	InvalidTimeStamp.Expired	Specified time stamp or date value is expired.	400	检查时间戳。
SignatureNonce 重复	SignatureNonceUsed	Specified signature nonce was used already.	400	

返回值格式不正确。	InvalidParameter.Format	Specified parameter format is not valid.	400	仅支持 XML/JSON。
参数值校验不通过。	Invalid{ParameterName}	Specified parameter {ParameterName} is not valid.	400	请检查指定参数的值。
Http 请求方法不支持。	UnsupportedHTTPMethod	Specified signature is not matched with our calculation.	400	请检查请求 Method。
签名方法不支持。	InvalidSignatureMethod	Specified signature method is not valid.	400	默认可以不填写该值。
签名不通过。	SignatureDoesNotMatch	Specified signature is not matched with our calculation.	400	签名不通过。
用户调用频率超限。	Throttling.User	Request was denied due to user flow control.	400	请稍后访问，降低访问频率。
API 访问频率超限。	Throttling.API	Request was denied due to api flow control.	400	请稍后访问，降低访问频率。
缺少AccessKeyId。	MissingSecurityToken	SecurityToken is mandatory for this action.	400	请检查是否传入有效AccessKeyId。

4.Invalid Signature

错误原因

客户端与服务端计算的签名不匹配导致的。

解决方案

当签名不匹配时网关会通过 HTTP Response Header 中的 X-Ca-Error-Message 返回服务端参与签名计算的 StringToSign。

StringToSign是用户请求前需要拼接的一个用于计算签名的字符串，在文档：【使用摘要签名认证方式调用API】查看详细说明。

客户端只需打印出本地自己拼接的 StringToSign 进行对比，找出哪里不同，针对性的解决，如果使用的官方提供的调用 Demo，可以到签名计算的工具类中找出计算签名前的 StringToSign 打印出来即可进行对比。

因为 HTTP Response Header 中不允许出现换行符，因此返回结果中的 StringToSign 换行符都已经被抹去。请参照文档合理比对。

如果服务端 StringToSign 与客户端一致，请再检查使用的 AppKey、AppSecret 是否正确，尤其注意是否额外错误添加了空格等不容易发现的字符。

5.Invalid Url

错误原因

请求传入的 HTTP Method 或者 Path 不正确，或者请求指定的环境（X-Ca-Stage）不正确。

如指定调用 TEST 环境的 API，但 API 并未被发布到测试。

注意：

- 请求时不指定环境，默认为访问 RELEASE 环境。
- 对 API 定义有修改，需要重新发布才能生效。许多出现这个错误的都是因为修改了 Path 没发布不生效，用新 Path 请求报错。

解决办法

分别检查上述三个因素：HTTP Method、Path、环境。

1. API 说明中要求用 POST 则不能用 GET 请求。Method 要一致。
2. Path 要与当前运行的一致。开放 API 的用户经常修改之后不发布，导致调用失败。
3. 指定环境要合理。在请求的 Header 里有个参数 X-Ca-Stage，取值 TEST/PRE/RELEASE，分别指向测试和线上环境，不传入该参数则默认是线上。

6.Unauthorized

错误原因

请求 API 时，使用的 AppKey 所属于的 APP 未获得授权。无权调用 API。

解决方法

授权生效的决定因素有：APP、API、环境、已授权

1. 如果是开放 API 的用户自己测试，则需要在 API 网关控制台，真实创建 APP，然后根据 AppId 在 API 列表页操作授权。即开放 API 的用户自测时，其实是自己需要给自己创建的 APP 授权。
2. 如果是购买了 API 的用户，则在 APP 详情页可以查看该 APP 已经被授权的 API，若没有要调用的 API，则自行操作授权。
3. 如果是使用了合作伙伴的 API，没有购买行为的。则联系合作伙伴。您需要提供 AppId，然后由 API 提供者操作授权。
4. 授权关系是有环境属性的，即 APP、API都是同一个的情况下，授权的环境和请求的环境也要相同。授权了某 API 在 A 环境的权限，但是也不能调用该 API 在 B 环境中的服务。
5. 最重要的一点是，确认清楚是否用错了 APP，是否调用错了 API。由于 API 和 APP 较多，很多用户因为搞混了而没能调用成功。比如授权了 APP A，但是调用时候用的是 APP B，请仔细排查。

7.ApiGateway服务关联角色

本文为您介绍ApiGateway服务关联角色（AliyunServiceRoleForApiGateway）的应用场景以及如何删除服务关联角色。

背景信息

ApiGateway服务关联角色（AliyunServiceRoleForApiGateway）是用户需要配置API后端为函数计算的情况下，为获取访问用户函数计算的访问权限，而提供的RAM角色。更多关于服务关联角色的信息请参见服务关联角色。

应用场景

API网关访问用户的函数计算，通过服务关联角色功能获取访问权限。

AliyunServiceRoleForApiGateway介绍

角色名称：AliyunServiceRoleForApiGateway 角色权限策略：AliyunServiceRolePolicyForApiGateway 权限说明：

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "fc:InvokeFunction"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

删除服务关联角色

如果您需要删除AliyunServiceRoleForApiGateway（服务关联角色），需要先删除后端为函数计算的API。

8.API网关 - 监控服务关联角色

本文为您介绍API网关 - 监控服务关联角色（AliyunServiceRoleForApiGatewayMonitoring）的应用场景以及如何删除API网关 - 监控服务关联角色。

背景信息

API网关 - 监控服务关联角色（AliyunServiceRoleForApiGatewayMonitoring）是为了方便用户对同一分组上的所有API应用相同的报警配置时，而提供的RAM角色。更多关于服务关联角色的信息请参见服务关联角色。

应用场景

API网关通过服务关联角色功能获取用户云监控资源访问权限，可以把API分组和云监控的应用分组建立起对应关系，使得指定API分组上的API会自动成为对应的云监控应用分组下的实例，从而批量管理同一分组下的API报警。

AliyunServiceRoleForApiGatewayMonitoring介绍

角色名称：AliyunServiceRoleForApiGatewayMonitoring 角色权限策略：AliyunServiceRolePolicyForApiGatewayMonitoring 权限说明：

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cms:CreateMonitorGroup",
        "cms:CreateMonitorGroupInstances",
        "cms:DescribeMonitorGroups",
        "cms:DeleteMonitorGroup",
        "cms:DeleteMonitorGroupInstances"
      ],
      "Resource": "acs:cms:*:*:*"
    }
  ]
}
```

删除服务关联角色

如果您需要删除AliyunServiceRoleForApiGatewayMonitoring（服务关联角色），需要先删除依赖这个服务关联角色的云监控下的应用分组。操作步骤如下：

1. 进入云监控控制台。
2. 点击左侧菜单 `应用分组` 。
3. 找到并删除名称为APIGATEWAY_${region}_$(groupId)的应用分组，region 表示云监控应用分组所对应的API分组所在的区域，groupId表示API分组ID。如：

   `APIGATEWAY_ap-southeast-1_d17e78c9d436436c89a7e8c42a329a4d`

4. 进入RAM控制台，点击左侧菜单 `RAM 角色管理` ，找到角色 `AliyunServiceRoleForApiGatewayMonitoring` 删除。

删除服务关联角色具体操作请参见删除服务关联角色。

9.后端服务调不通

- 需要检查您录入的后端服务地址是否正确。
- 需要保证您的后端服务可以被网关访问到。
- 如果使用的是内网 IP ，请保证您的后端服务于您的 API 处于同一地域。
- 检查您API定义的“后端超时”时间。
- 该文档仅针对经典网络内网（后端是经典网络的ECS/SLB）或者公网访问后端服务，如果后端ECS在VPC内，请参考VPC的配置页面创建后端服务为VPC内资源的API。

  在API定义时会要求您录入一个超时时间，当您的后端服务没有在您指定的时间内返回时，API网关仍然会提示您无法链接后端服务。您可以根据后端服务的实际耗时对"后端超时"进行调整，最大支持30000ms。

  注意：单位是ms（毫秒）。

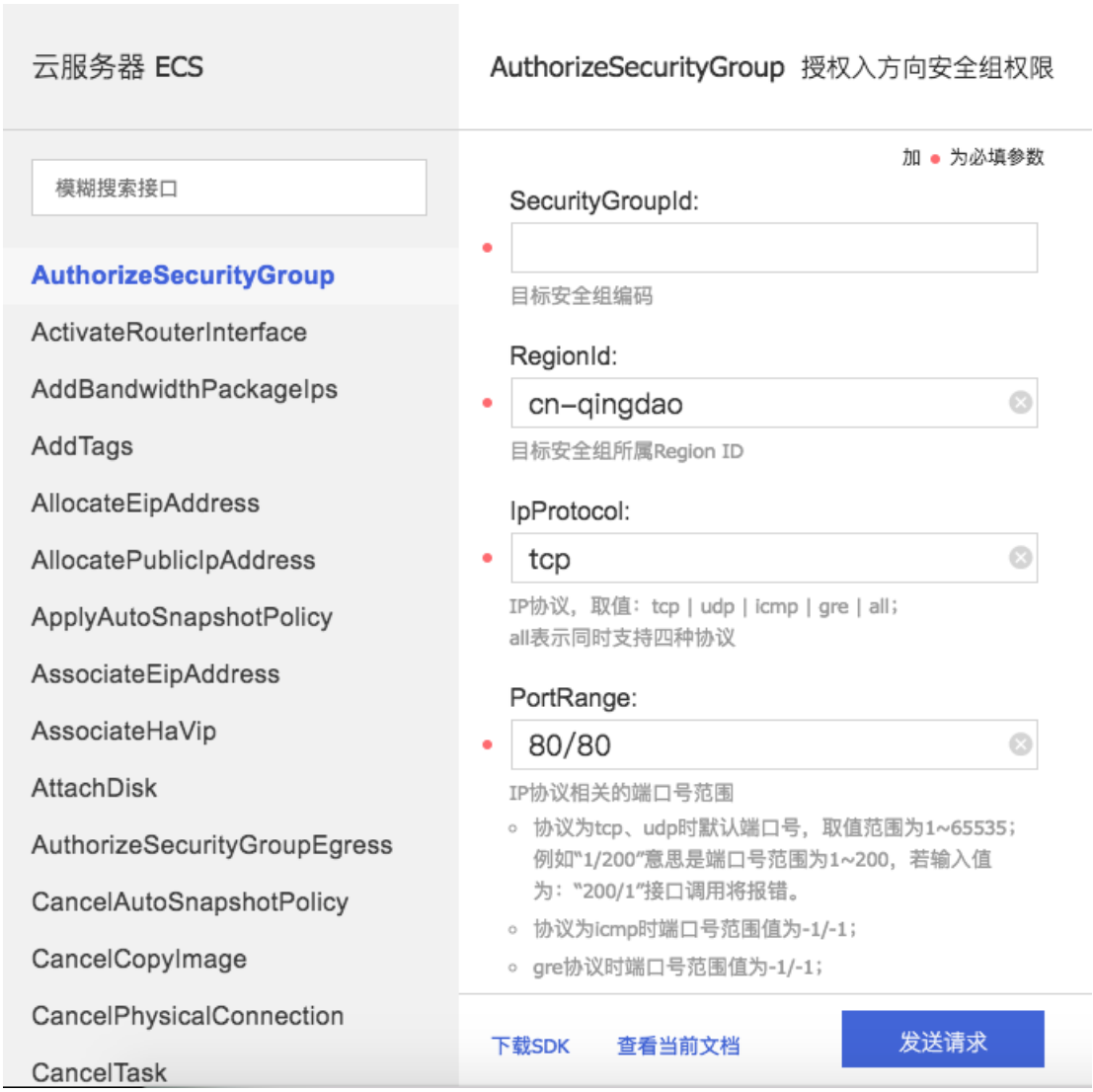
- 如后端服务在 ECS ，请检查安全组设置，是否可以被外部访问。请保证安全组可以被API网关所在的IP段访问。

需要强调，API网关访问外部的出口IP不能保证不会变动，通过设置安全组来判断请求来源的方法可能会由于API网关出口IP变化而导致API网关访问后端服务失败，因此此我们不建议使用安全组方法。API网关提供了完整的后端签名验证的方式来做身份认证，使用后端签名机制可以完全避免这种不确定性，我们强烈建议此方法来确认请求来源。使用具体文档：后端签名密钥说明文档。

* **设置方法**

点击链接：设置ECS安全组

跳转至：



在发送调用请求前，请调整如下配置项，其他项为选填，如无特殊要求，请不要改动。

```
* RegionId，区域ID，请填写资源所在的区域ID，例如：cn-shenzhen、cn-shanghai
* PortRange：端口，若后端服务HTTP为：80/80，HTTPS：443/443
* NicType：网络类型，取值：
  * internet：外网
  * intranet：内网
  * 若以如上述链接进入，则已经为您默认设置了：intranet（内网）
* SourceCidrIp:允许的源IP地址范围，以下地址为API网关的出口IP，请区分区域填写<br>
```

出口IP查看方法：

API网关的出口IP请查看分组所在实例的出口IP，具体查看方法： 先通过API网关控制台-【开放API】-【分组管理】-【分组详情】查看分组所在实例信息。

然后到【实例】页面查看对应实例的出口地址。

填写完成后，点击**发送请求** ，完成设置。

当之返回一个RequestID时，表示设置成功。

确认添加是否成功。

可以到控制台-【ECS控制台】-【安全组】-找到相应安全组，点击查看“安全组规则”的“内网入方向”或者“公网入方向”。

10.API定义

10.1. API网关是否支持gzip数据压缩，是否需要添加请求头Accept-Encoding ？

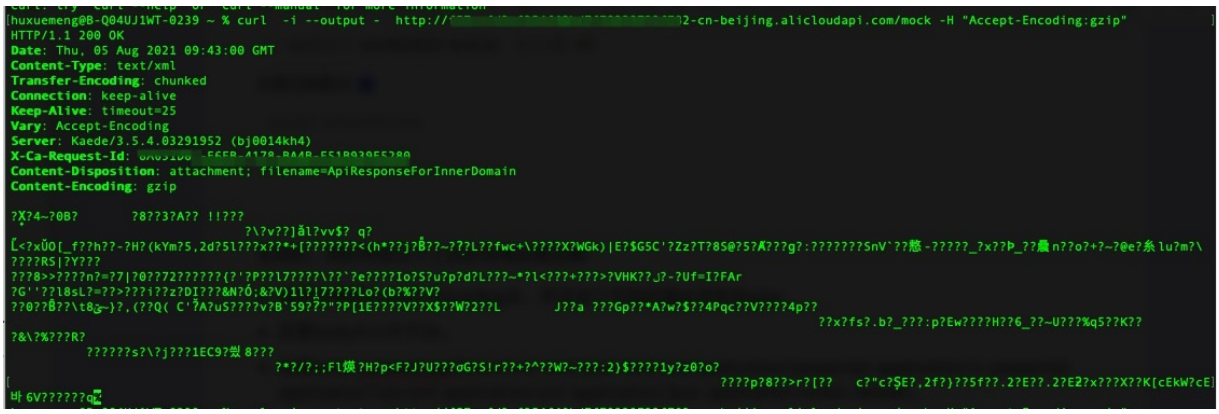
支持的，同时符合以下三点时会将应答压缩：

- 请求中携带Accept-Encoding头，并且这个字段的值必须包含gzip。
- 应答body大小大于2k。
- 应答body Content-Type 为text/xml text/plain text/css application/javascript application/x-javascript application/rss+xml application/xml application/json application/octet-stream。

示例：

配置了一个后端为mock的API，应答的Content-Type为text/xml ，应答大小为6k。测试如下：

同时满足以上三个条件时，调用API收到的应答是压缩后的数据，需客户端自行解压。



11.调用API

11.1. 请求中编码问题

API请求编码问题

客户端请求API网关时，需要对参数值进行utf-8的urlEncode，这样能避免特殊参数或者中文出现乱码。 注意：query、header、body位置编码需要在签名计算后，签名时用原始值。

以下代码可参考：https://github.com/aliyun/api-gateway-demo-sign-java。

1、header值进行编码

header的编码和value编码不太一样，header的值需要用ISO-9959-1编码。

```
HttpPost post = new HttpPost(initUrl(host, path, querys));
for (Map.Entry<String, String> e : headers.entrySet()) {
 //header的传值，value要编码后传，具体实现请看下面
 post.addHeader(e.getKey(),  MessageDigestUtil.utf8ToIso88591(e.getValue()));
 }
```

```
/**
* UTF-8编码转换为ISO-9959-1     *
* @param str
* @return
*/
public static String utf8ToIso88591(String str) {
if (str == null) {
            return str;
}
try {
             return new String(str.getBytes("UTF-8"), "ISO-8859-1");         }
catch (UnsupportedEncodingException e) {
             throw new RuntimeException(e.getMessage(), e);
        }
}
```

2、query值进行编码

如果query参数值含中文，需要对query的值进行utf-8编码。可参考下面url的构建方式。

```
private static String initUrl(String host, String path, Map<String, String> querys) throws
UnsupportedEncodingException {
     StringBuilder sbUrl = new StringBuilder();
     sbUrl.append(host);
     if (!StringUtils.isBlank(path)) {
      sbUrl.append(path);
        }
     if (null != querys) {
      StringBuilder sbQuery = new StringBuilder();
         for (Map.Entry<String, String> query : querys.entrySet()) {
          if (0 < sbQuery.length()) {
           sbQuery.append("&");
          }
          if (StringUtils.isBlank(query.getKey()) && !StringUtils.isBlank(query.getValue())
) {
           sbQuery.append(query.getValue());
                }
          if (!StringUtils.isBlank(query.getKey())) {
           sbQuery.append(query.getKey());
           if (!StringUtils.isBlank(query.getValue())) {
            sbQuery.append("=");
            sbQuery.append(URLEncoder.encode(query.getValue(), "UTF-8"));
           }
                }
         }
         if (0 < sbQuery.length()) {
          sbUrl.append("?").append(sbQuery);
         }
        }
     return sbUrl.toString();
}
```

3、body参数进行编码

1）form形式body的编码:

```
UrlEncodedFormEntity formEntity = buildFormEntity(bodys);
    if (formEntity != null) {
        post.setEntity(formEntity);
    }
```

```
 /**
     * 构建FormEntity
     *
     * @param formParam
     * @return
     * @throws UnsupportedEncodingException
     */
    private static UrlEncodedFormEntity buildFormEntity(Map<String, String> formParam)
            throws UnsupportedEncodingException {
        if (formParam != null) {
            List<NameValuePair> nameValuePairList = new ArrayList<NameValuePair>();
            for (String key : formParam.keySet()) {
                nameValuePairList.add(new BasicNameValuePair(key, formParam.get(key)));
            }
            UrlEncodedFormEntity formEntity = new UrlEncodedFormEntity(nameValuePairList, "
UTF-8");
            formEntity.setContentType("application/x-www-form-urlencoded; charset=UTF-8");
            return formEntity;
        }
        return null;
    }
```

2）非form形式的body

String 形式的body:

```
if (StringUtils.isNotBlank(body)) {
    post.setEntity(new StringEntity(body, "UTF-8"));
}
```

byte[]形式的body:

```
if (bodys != null) {
    post.setEntity(new ByteArrayEntity(bodys));
}
```

4、path参数编码

当API的path中有参数，而且参数含特殊字符。需要先将参数urlencode后加入path中。不然签名会有问题。此时后端收到的也是urlEncode后的值，如果要获取原值，需要自己在后端进行urlDecode。

```
@Test
    public void testPath() throws Exception{
        //请求path
        String host="你的域名";
        //请求path，先将参数处理后再放入path中。
        String type="中文 123";
        String pathParam= URLEncoder.encode(type,"UTF-8");
        String path = "/"+pathParam;
        Map<String, String> headers = new HashMap<String, String>();
        //（必填）根据期望的Response内容类型设置
        headers.put(HttpHeader.HTTP_HEADER_ACCEPT, "application/json");
        CUSTOM_HEADERS_TO_SIGN_PREFIX.clear();
        Request request = new Request(Method.GET, host, path, AppKey, AppSecret, Constants.
DEFAULT_TIMEOUT);
        request.setHeaders(headers);
        request.setSignHeaderPrefixList(CUSTOM_HEADERS_TO_SIGN_PREFIX);
        //调用服务端
        Response response = Client.execute(request);
        System.out.println(JSON.toJSONString(response));
    }
```

5、参数中包含emoji表情

当参数中包含emoji表情，需要先对参数值进行urlencode后再签名，不然签名不过，因为API网关的系统可能会识别不了该符号，会导致签名失败，所以需要先进行urlencode处理后再签名。

此时后端收到的为urlencode后的字符串，如果要获取原文，后端需要自己做urldecode。