

API 网关

常见问题

常见问题

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：

```
X-Ca-Request-Id: 7AD052CB-EE8B-4DFD-BBAF-EFB340E0A5AF
//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-Error-Message: Invalid Url
//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-Debug-Info: {"ServiceLatency":0,"TotalLatency":2}
//当打开Debug模式后会返回Debug信息，此信息后期可能会有变更，仅用做联调阶段参考
```

在 Header 中获得 X-Ca-Error-Message 可以基本明确报错原因，而 X-Ca-Request-Id 可以用于提供给这边的支持人员，供支持人员搜索日志。

返回值为空

HTTP/HTTPS 请求的返回结果有 HttpStatusCode、Header、Body 三部分。当请求报错时，由于没有进入业务逻辑，所以返回的 Body 可能为空，表现为“返回值为空”，但实际上，重要信息都在 Header 里面。

用户发起的 API 请求只要能够到达网关，就会返回成功/错误的结果信息。

重要的返回信息都在Header里面，以X-Ca开头的为网关返回的信息。其中较主要的为下面的几个信息：

```
X-Ca-Request-Id: 7AD052CB-EE8B-4DFD-BBAF-EFB340E0A5AF
//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-Error-Message: Invalid Url
```

```
//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-Debug-Info: {"ServiceLatency":0,"TotalLatency":2}
```

```
//当打开Debug模式后会返回Debug信息，此信息后期可能会有变更，仅用做联调阶段参考
```

所以如果发送请求后，发现返回值为空，那么看一下返回的 Header 信息。如果请求到达网关就错误返回了，那么 Body 为空很正常，会表现为返回值为空，但是在 Header 里面会有重要信息。

如果Header也为空，那么说明请求没有达到网关，请自行检查网络状况等。

各种语言获取和查看 HTTP/HTTPS 头部信息的方法均可在网上查询到。

HTTPS 证书报错

错误提示

调用 HTTPS 接口出现证书认证错误或者提示证书已经过期。

原因及解决方案

1. 证书不合法

API 提供者使用的证书其他非主流机构颁发的证书，此类证书使用浏览器访问是没有问题的，因为浏览器会自动更新根证书。但老版本操作系统的根证书对这些机构（证书颁发机构）的不信任或者信任时间已经过期。

解决方案

1. 升级客户端根证书。如：Java+Linux 升级 OpenSSL 客户端即可，其他操作系统+编程语言，请升级编程语言中 HTTPS 使用的根证书。
2. 联系 API 提供者，要求其更换兼容性更好的主流 SSL 证书。
3. 程序中忽略检查 SSL 证书有效性（不推荐），忽略后会有请求被劫持的安全风险。只在与 API 提供商无法提供兼容性更好的主流 SSL 证书，且安全风险可控的前提下，可以考虑使用此方法。

2. API提供者的 SSL 证书过期

提供者的 SSL 证书过期。

解决方案

1. 联系 API 提供者，要求其更换 SSL 证书。

2. 程序中忽略检查 SSL 证书有效性（不推荐），忽略后会有请求被劫持的安全风险。只在与 API 提供商无法提供兼容性更好的主流 SSL 证书，且安全风险可控的前提下，可以考虑使用此方法。

错误代码表

当您调用您或者他人在API网关开放的API服务时，如天气查询、IP查询、图像识别等API服务，您可能遇到以下这些错误码。

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。

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。
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.请见使用限制
Quota Exhausted	403	调用次数已用完	购买的次数已用完。
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。请参照错误说明 Invalid Url。
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，参照 Invalid Signature 错误说明
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。请参照 请求签名说明文档。

			档。
Unauthorized	403	未被授权	APP 未获得要调用的 API 的授权。请参照错误说明 Unauthorized。
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。

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。

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。

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	解决方案
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	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

	license, or the ICP license does not belong to Aliyun.			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。
ServiceInDebt	Your API Gateway	400	（您的 API 网关）服务欠费	充值或者结算后重新开始使用。

	service is in dept.			
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。
CertificateNot Match	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。

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	404	请检查调用时是否传入正确的

		is not found.		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	SignatureNonce Used	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 {Parameter Name} 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。	InvalidSignature Method	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	400	签名不通过。

		with our calculation .		
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。

API unsupport the channel

错误原因：

请求的 HTTP Schema 不正确。

不同 API 支持的 HTTP Schema 不同，API 提供者可以自行设置，支持仅 HTTP 访问、仅 HTTPS 访问，或者同时支持 HTTP 和 HTTPS。

API unsupport the channel: HTTP，API 只允许 HTTPS 访问，而发起 API 请求时却使用了 HTTP，则会得到此提示。

API unsupport the channel: HTTPS，API 只允许 HTTP 访问，而 API 请求却使用的是 HTTPS，则会得到此提示。

解决方案：

API unsupport the channel: HTTP，更换成为 HTTPS，发起 API 调用。

API unsupport the channel: HTTPS，更换成为 HTTP，发起 API 调用。

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：【请求签名说明文档】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。

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。
4. 更多参数及请求说明，参见 API 调用示例

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。请求时的环境等参数指定，请参见【API 请求示例】。
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。

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与您的 API 处于同一地域。

检查您API定义的“后端超时”时间。

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）

API定义

授权信息

发布历史

监控信息

安全信息

调试API

后端基础定义

协议 ☒ HTTP/HTTPS

后端服务地址

后端服务地址指API网关调用底层服务时的域名或者IP，不包含Path
[为什么无法调通我的后端服务？](#)

后端请求Path

后端请求Path必须包含后端服务参数中的Parameter Path，包含在[]中，比如

HTTP Method GET

后端超时 100 ms

Mock 不使用Mock

- 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。

- 设置方法点击链接：设置ECS安全组
跳转至：

云服务器 ECS

模糊搜索接口

AuthorizeSecurityGroup

ActivateRouterInterface

AddBandwidthPackageIds

AddTags

AllocateEipAddress

AllocatePublicIpAddress

ApplyAutoSnapshotPolicy

AssociateEipAddress

AssociateHaVip

AttachDisk

AuthorizeSecurityGroupEgress

CancelAutoSnapshotPolicy

CancelCopyImage

CancelPhysicalConnection

CancelTask

AuthorizeSecurityGroup 授权入方向安全组权限

加 为必填参数

SecurityGroupId:

目标安全组编码

RegionId:

cn-qingdao

目标安全组所属Region ID

IpProtocol:

tcp

IP协议，取值：tcp | udp | icmp | gre | all；
all表示同时支持四种协议

PortRange:

80/80

IP协议相关的端口号范围

- 协议为tcp、udp时默认端口号，取值范围为1~65535；
例如“1/200”意思是端口号范围为1~200，若输入值为：“200/1”接口调用将报错。
- 协议为icmp时端口号范围值为-1/-1；
- gre协议时端口号范围值为-1/-1；

下载SDK

查看当前文档

发送请求

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

注：部分区域的API网关存在多个出口IP段，均需要加入安全组（每次操作只能添加一个IP段，多个IP段的区域需重复执行操作），添加时请注意内、外网。

区域	内网	外网
华北 1（青岛）	10.151.203.0/24	
华北 2（北京）	10.152.167.0/24	123.56.213.0/24
华南 1（深圳）	10.152.27.0/24	120.76.91.0/24
华东 1（杭州）	10.152.29.0/24 10.152.30.0/24	114.55.70.0/24
华东 2（上海）	10.152.163.0/24 10.152.164.0/24 11.192.97.0/24	139.224.92.0/24 139.224.92.0/24 139.224.4.0/24

- 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

内网入方向

内网出方向

公网入方向

公网出方向

经典网络的内网入方向规则，推荐优先选择安全组授权方式，如选择IP地址方式授权，出于安全性的考虑，仅支持单IP授权，例如：10.x.y.z/32。 [教我设置](#)

授权策略	协议类型	端口范围	授权类型	授权对象	描述	优先级	操作
允许	自定义 TCP	80/80	地址段访问	10.152.29.0/24	-	3	克隆 删除

API安全

API定义

产品说明

调用API

POST上传文件

POST调用——文件上传

由于API网关暂不支持multipart形式的文件上传，所以需要客户端调用时将文件内容转化为二进制的Base64编码，传给网关。后端收到数据内容，也需要进行相应的Base64解码获得文件流。

1、API配置

一般包含文件上传的API接口，建议配置为“非Form表单”。非Form表单的数据不会参与签名，而是通过content-md5去防数据篡改。

2、Content-Type

1) body内容为一个JSON格式内容，Content-Type可配置为“application/json; charset=UTF-8”。

- 调用示例

如调用API 印刷文字识别_身份证识别。

body内容为一个JSON格式，其中的参数dataValue为图片内容，传输时需要将图片内容转化为二进制的Base64编码。

```

    public static void main(String[] args) {
//请求path
String host = "https://dm-51.data.aliyun.com";
String path = "/rest/160601/ocr/ocr_idcard.json";
String appKey="你自己的AppKey";
String appSecret="你自己的AppSecret";

//图片转化为二进制的Base64编码字符串
String imgFile="/Users/wuling/Downloads/dm-51示例图片.jpg";
InputStream in = null;
byte[] data = null;
//读取图片字节数组
try {
in = new FileInputStream(imgFile);
data = new byte[in.available()];
in.read(data);
in.close();
} catch (IOException e) {
e.printStackTrace();
}
String base64String=new String(Base64.encodeBase64(data));
//Body内容
String body = "{\"inputs\":{\"image\":{\"dataType\":\"50\",\"dataValue\":\"\" + base64String
+ \"\"},\"configure\":{\"dataType\":\"50\",\"dataValue\":\"{\\\"side\\\":\\\"face\\\"}\"}}}\"";

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");
// ( 可选 ) Body MD5,服务端会校验Body内容是否被篡改,建议Body非Form表单时添加此Header
headers.put(HttpHeader.HTTP_HEADER_CONTENT_MD5, MessageDigestUtil.base64AndMD5(body));
// ( POST/PUT请求必选 ) 请求Body内容格式
headers.put(HttpHeader.HTTP_HEADER_CONTENT_TYPE, "application/json; charset=UTF-8");

Request request = new Request(Method.POST_STRING, host, path, appKey, appSecret,
Constants.DEFAULT_TIMEOUT);
request.setHeaders(headers);
request.setSignHeaderPrefixList(CUSTOM_HEADERS_TO_SIGN_PREFIX);

request.setStringBody(body);
/**
 * 重要提示如下:
 * 代码中用到的类请下载 https://github.com/aliyun/api-gateway-demo-sign-java
 *
 * 具体调用请参考https://github.com/aliyun/api-gateway-demo-sign-java/blob/master/src/test/java/com/aliyun/api/gateway/demo/Demo.java
 */
//调用服务端
Response response = Client.execute(request);

System.out.println(JSON.toJSONString(response));
}

```

2) body内容仅为文件内容，Content-Type可配置为“application/octet-stream; charset=UTF-8”。

- 调用示例

如调用API 九云图文档转 H5 (SVG) 网页-文档转换POST。

```
public static void main(String[] args) {
//请求path
String host = "https://api.9yuntu.cn";
String path = "/execute/PostConvert";
String appKey="你自己的AppKey";
String appSecret="你自己的AppSecret";

//文件转化为二进制的Base64编码字符串
String imgFile="/Users/wuling/Downloads/demo.docx";
InputStream in = null;
byte[] data = null;
//读取字节数组
try {
in = new FileInputStream(imgFile);
data = new byte[in.available()];
in.read(data);
in.close();
} catch (IOException e) {
e.printStackTrace();
}

String body=new String(Base64.encodeBase64(data));

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");
// ( 可选 ) Body MD5,服务端会校验Body内容是否被篡改,建议Body非Form表单时添加此Header
headers.put(HttpHeader.HTTP_HEADER_CONTENT_MD5, MessageDigestUtil.base64AndMD5(body));
// ( POST/PUT请求必选 ) 请求Body内容格式
headers.put(HttpHeader.HTTP_HEADER_CONTENT_TYPE, "application/octet-stream; charset=UTF-8");

Request request = new Request(Method.POST_STRING, host, path, appKey, appSecret,
Constants.DEFAULT_TIMEOUT);
request.setHeaders(headers);
request.setSignHeaderPrefixList(CUSTOM_HEADERS_TO_SIGN_PREFIX);

request.setStringBody(body);

//请求的query
Map<String, String> querys = new HashMap<String, String>();
querys.put("docName", "demo");
querys.put("outputType", "html");
request.setQuerys(querys);
/**
 * 重要提示如下:
 * 代码中用到的常量请下载 https://github.com/aliyun/api-gateway-demo-sign-java
 *
 * 具体调用请参考https://github.com/aliyun/api-gateway-demo-sign-java/blob/master/src/test/java/com/aliyun/api/gateway/demo/Demo.java
 */
//调用服务端
Response response = Client.execute(request);
```

```
System.out.println(JSON.toJSONString(response));
}
```

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 UnsupportedOperationException
*/
private static UrlEncodedFormEntity buildFormEntity(Map<String, String> formParam)
throws UnsupportedOperationException {
    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。

分组相关

监控与报警

相关协议

域名相关