阿里云
密钥管理服务

API参考

文档版本：20190916

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
	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。	 **禁止:** 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	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。	单击 **确定**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
*## *	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。	`swich` *`{stand | slave}`*

目录

1 调用方式

1.1 概述

对 KMS API 接口调用是通过向 KMS API 的服务端地址发送 HTTP POST 和 GET 请求，并按照接口说明在请求中加入相应请求参数来完成的；根据请求的处理情况，系统会返回处理结果。

1. #unique_5
2. #unique_6
3. #unique_7
4. #unique_8

1.2 请求结构

本文为您介绍KMS API的服务接入地址、通信协议、HTTP请求方法和请求参数等信息。

服务地址

KMS API的服务接入地址，如下表所示。

地域	RegionId	公网接入地址	VPC接入地址
亚太东北1（东京）	ap-northeast-1	kms.ap-northeast-1.aliyuncs.com	kms-vpc.ap-northeast-1.aliyuncs.com
亚太东南1（新加坡）	ap-southeast-1	kms.ap-southeast-1.aliyuncs.com	kms-vpc.ap-southeast-1.aliyuncs.com
亚太东南2（悉尼）	ap-southeast-2	kms.ap-southeast-2.aliyuncs.com	kms-vpc.ap-southeast-2.aliyuncs.com
亚太东南3（吉隆坡）	ap-southeast-3	kms.ap-southeast-3.aliyuncs.com	kms-vpc.ap-southeast-3.aliyuncs.com
亚太东南5（雅加达）	ap-southeast-5	kms.ap-southeast-5.aliyuncs.com	kms-vpc.ap-southeast-5.aliyuncs.com
亚太南部1（孟买）	ap-south-1	kms.ap-south-1.aliyuncs.com	kms-vpc.ap-south-1.aliyuncs.com

地域	RegionId	公网接入地址	VPC接入地址
华东1（杭州）	cn-hangzhou	kms.cn-hangzhou.aliyuncs.com	kms-vpc.cn-hangzhou.aliyuncs.com
华东2（上海）	cn-shanghai	kms.cn-shanghai.aliyuncs.com	kms-vpc.cn-shanghai.aliyuncs.com
华北1（青岛）	cn-qingdao	kms.cn-qingdao.aliyuncs.com	kms-vpc.cn-qingdao.aliyuncs.com
华北2（北京）	cn-beijing	kms.cn-beijing.aliyuncs.com	kms-vpc.cn-beijing.aliyuncs.com
华北3（张家口）	cn-zhangjiakou	kms.cn-zhangjiakou.aliyuncs.com	kms-vpc.cn-zhangjiakou.aliyuncs.com
华北5（呼和浩特）	cn-huhehaote	kms.cn-huhehaote.aliyuncs.com	kms-vpc.cn-huhehaote.aliyuncs.com
华南1（深圳）	cn-shenzhen	kms.cn-shenzhen.aliyuncs.com	kms-vpc.cn-shenzhen.aliyuncs.com
欧洲中部1（法兰克福）	eu-central-1	kms.eu-central-1.aliyuncs.com	kms-vpc.eu-central-1.aliyuncs.com
中东东部1（迪拜）	me-east-1	kms.me-east-1.aliyuncs.com	kms-vpc.me-east-1.aliyuncs.com
中国香港（香港）	cn-hongkong	kms.cn-hongkong.aliyuncs.com	kms-vpc.cn-hongkong.aliyuncs.com
美国东部1（弗吉尼亚）	us-east-1	kms.us-east-1.aliyuncs.com	kms-vpc.us-east-1.aliyuncs.com
美国西部1（硅谷）	us-west-1	kms.us-west-1.aliyuncs.com	kms-vpc.us-west-1.aliyuncs.com
华东 1（杭州金融云）	cn-hangzhou-finance	kms.cn-hangzhou-finance.aliyuncs.com	无
华东 2（上海金融云）	cn-shanghai-finance-1	kms.cn-shanghai-finance-1.aliyuncs.com	kms-vpc.cn-shanghai-finance-1.aliyuncs.com

地域	RegionId	公网接入地址	VPC接入地址
华南 1（深圳金融云）	cn-shenzhen-finance-1	kms.cn-shenzhen-finance-1.aliyuncs.com	kms-vpc.cn-shenzhen-finance-1.aliyuncs.com
英国（伦敦）	eu-west-1	kms.eu-west-1.aliyuncs.com	kms-vpc.eu-west-1.aliyuncs.com

通信协议

KMS服务只支持使用HTTPS通道发送请求。

KMS不支持SSLv2和SSLv3，仅支持TLS1.0及以上。

请求方法

支持HTTP POST和GET方法发送请求，这种方式下请求参数需要包含在请求的URL中。

请求参数

每个请求都需要指定要执行的操作，即`Action`参数（例如 #unique_10），以及每个操作都需要包含的#unique_6 和指定操作所特有的请求参数。

字符编码

请求及返回结果都使用`UTF-8`字符集编码。

1.3 公共参数

公共请求参数是指每个接口都需要使用到的请求参数。

公共请求参数

名称	类型	是否必须	描述
Format	String	否	返回值的类型，支持 JSON 与 XML。默认为 XML。
Version	String	是	API 版本号，为日期形式：YYYY-MM-DD，本版本对应为 2016-01-20。
AccessKeyId	String	是	阿里云颁发给用户的访问服务所用的密钥 ID。

名称	类型	是否必须	描述
Signature	String	是	签名结果串。关于签名的计算方法，请参见签名机制。
SignatureMethod	String	是	签名方式，目前支持HMAC-SHA1。
Timestamp	String	是	请求的时间戳。日期格式按照 ISO8601 标准表示，并需要使用 UTC 时间。格式为：YYYY-MM-DDThh:mm:ssZ。例如，2015-12-01T12:00:00Z。
SignatureVersion	String	是	签名算法版本，目前版本是 1.0。

示例

```
https://kms.cn-hangzhou.aliyuncs.com/?
Format=json
&Version=2016-01-20
&AccessKeyId=testid
&Signature=YlrFhyqDZQ1ThNYARrv3Ptaxqf****
&SignatureMethod=HMAC-SHA1
&Timestamp=2016-03-25T09:36:58Z
&SignatureVersion=1.0
```

公共返回参数

用户发送的每次接口调用请求，无论成功与否，系统都会返回一个唯一识别码 RequestId 给用户。

示例

XML示例

```
<KMS>
<RequestId>348d9445-e39a-4d80-907d-298cc6c94447</RequestId>
<!—返回结果数据-->
</KMS>
```

JSON示例

```
{
"RequestId": "284b2b80-9b17-4546-a093-adfbae512a54"
}
```

1.4 签名机制

用户在 HTTP 或 HTTPS 请求中添加签名（Signature）信息，阿里云才可以对请求进行身份验证。签名需要使用 AccessKey，由 AccessKey ID 和 AccessKey Secret 组成。用户可通过阿里云官方网站申请和管理 AccessKey ID 和 AccessKey Secret。

签名步骤

1. 构造规范化的请求字符串

   - 按照字母升序，对参数名称进行排序。排序的参数包括公共请求参数和要调用的接口的指定参数。

     说明:
     当使用GET方法提交请求时，这些参数就是请求URI中的参数部分（即URI中“?”之后由“&”连接的部分）。

   - 对每个请求参数的名称和值进行URL编码。使用UTF-8字符集进行编码，编码规则是：
     - 对于字符 A-Z、a-z、0-9以及字符“-”、“_”、“.”、“~”不编码；
     - 对于其他字符编码成“%XY”的格式，其中XY是字符对应ASCII码的16进制表示。比如英文的双引号（"）对应的编码就是%22；
     - 英文空格（ ）编码为%20，而不是加号（+）。

       说明:
       一般支持URL编码的库（比如Java中的java.net.URLEncoder）都是按照“application/x-www-form-urlencoded”的MIME类型的规则进行编码的。实现时可以直接使用这类方式进行编码，把编码后的字符串中加号（+）替换成%20、星号（*）替换成%2A、%7E替换回波浪号（~），即可得到上述规则描述的编码字符串。

   - 对编码后的参数名称和值使用英文等号（=）进行连接。
   - 再把英文等号连接得到的字符串按参数名称的字典顺序依次使用 & 符号连接，即得到规范化请求字符串。

     使用上一步构造的规范化字符串按照下面的规则构造用于计算签名的字符串：

```
StringToSign=
HTTPMethod + “&” +
percentEncode(“/”) + ”&” +
```

```
percentEncode(CanonicalizedQueryString)
```

其中 HTTPMethod 是提交请求用的 HTTP 方法，比 GET。percentEncode(“/”)是按照前文描述的 URL 编码规则对字符“/”进行编码得到的值，即“%2F”。

percentEncode(CanonicalizedQueryString)是对第1步中构造的规范化请求字符串按 1.b 中描述的 URL 编码规则编码后得到的字符串。

2. 按照RFC2104的定义，使用上面的用于签名的字符串计算签名 HMAC 值。注意：计算签名时使用的 Key 就是用户持有的 Access Key Secret 并加上一个“&”字符(ASCII:38)，使用的哈希算法是 SHA1。
3. 按照 Base64 编码规则把上面的 HMAC 值编码成字符串，即得到签名值（Signature）。
4. 将得到的签名值作为 Signature 参数添加到请求参数中，即完成对请求签名的过程。

说明:

得到的签名值在作为最后的请求参数值提交给 KMS 服务器的时候，要和其他参数一样，按照 RFC3986 的规则进行 URL 编码）。

示例

以`CreateKey`为例,签名前的请求 URL 为:

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=CreateKey
&SignatureVersion=1.0
&Format=json
&Version=2016-01-20
&AccessKeyId=testid
&SignatureMethod=HMAC-SHA1
```

```
&Timestamp=2016-03-28T03:13:08Z
```

那么CanonicalizedQueryString就是：

```
AccessKeyId=testid&Action=CreateKey&Format=json&SignatureMethod=HMAC-
SHA1&SignatureVersion=1.0&Timestamp=2016-03-28T03%3A13%3A08Z&Version=
2016-01-20
```

所以StringToSign应该是：

```
GET&%2F&AccessKeyId%3Dtestid&Action%3DCreateKey&Format%3Djson&
SignatureMethod%3DHMAC-SHA1&SignatureVersion%3D1.0&Timestamp%3D2016-03
-28T03%253A13%253A08Z&Version%3D2016-01-20
```

假如使用的 AccessKey Id 是testid，AccessKey Secret 是testsecret，用于计算 HMAC的 Key 就是testsecret&，则计算得到的签名值是：

```
s/OdVWMTmNGagvWlljdAJ7Itsew=
```

签名后的请求 URL 为（注意增加了Signature参数）：

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=CreateKey
&SignatureVersion=1.0
&Format=json
&Version=2016-01-20
&AccessKeyId=F585********APMU
&SignatureMethod=HMAC-SHA1
&Timestamp=2016-03-28T03:13:08Z
&Signature=41wk2SSX1GJh7fwnc5eqOfiJPF****
```

1.5 返回结果

您调用API服务后，系统会返回HTTP状态码。如果返回的状态码为2xx，代表调用成功；返回4xx或5xx代表调用失败。调用成功返回的数据格式暂时只支持JSON。如果您使用外部系统发送请求，可在参数中定制返回的数据格式。

成功结果

XML示例

```
<?xml version="1.0" encoding="UTF-8"?>
<!--结果的根结点-->
<接口名称+Response>
    <!--返回请求标签-->
    <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
    <!--返回结果数据-->
</接口名称+Response>
```

JSON示例

```
{
```

```
    "RequestId": "4C467B38-3910-447D-87BC-AC049166F216",
    /* 返回结果数据 */
}
```

错误结果

调用接口出错后，将不会返回结果数据。调用方可根据每个接口对应的错误码以及公共错误码来定位错误原因。

当调用出错时，HTTP请求返回一个4xx或5xx的HTTP状态码。返回的消息体中是具体的错误代码及错误信息。另外还包含一个全局唯一的请求 ID：RequestId和一个您该次请求访问的站点ID：HostId。在调用方找不到错误原因时，可以联系阿里云客服，并提供该HostId和RequestId，以便帮您尽快解决问题。

`XML`示例（请求过期）

```
<KMS>
<HttpStatus>400</HttpStatus>
<Code>IllegalTimestamp</Code>
<Message>The input parameter "Timestamp" that is mandatory for
processing this request is not supplied.</Message>
<RequestId>3b237773-bc2c-4bea-95fc-319a1a5baa68</RequestId>
</KMS>
```

`JSON`示例（请求过期）

```
{
"HttpStatus": 400
"Code": "IllegalTimestamp"
"Message": "The input parameter "Timestamp" that is mandatory for
processing this request is not supplied."
"RequestId": "e85db688-a2d3-44ca-9790-4259f59e90d8"
}
```

公共错误码

错误代码	描述	HTTP 状态码
InternalFailure	Internal Failure.	500
ServiceUnavailableTemporary	Service Unavailable Temporary.	503
InvalidAccessKeyId.NotFound	The AccessKey ID provided does not exist in our records.	404
Forbidden.KeyNotFound	The specified Key is not found.	404
Forbidden.KeyVersionNotFound	The specified Key version is not found.	404

错误代码	描述	HTTP 状态码
Forbidden.AliasNotFound	The specified Alias is not found.	404
Forbidden.NoPermission	This operation is forbidden by permission system.	403
Forbidden.AccessKey	This AccessKey is not enabled.	403
UnsupportedHTTPMethod	This http method is not supported.	403
Forbidden.UbsmsInvalidUserid	Userid Invalid For Ubsms.	403
Forbidden.UbsmsInvalidBid	Your account partner does not have KMS Service.	403
Forbidden.KmsServiceNotEnabled	Kms service is not Enabled for current user. Please get access permission first.	403
Forbidden.ProhibitedByRiskControl	Current user is Prohibited By Risk Control.	403
Forbidden.InDebtOverdue	Current user is indebted Overdue.	403
Forbidden.InDebt	Current user is indebted.	403
ParseRequestParameterException	Server parse parameters exception. Please check your input params.	400
MissingParameter	The parameter "< parameter name >" is needed but not provided.	400
InvalidParameter	The specified parameter "< parameter name >" is not valid.	400
IncompleteSignature	The request signature does not conform to Alibaba Cloud standards.	400
IllegalTimestamp	The input parameter "Timestamp" that is required for processing this request is not supplied .	400

错误代码	描述	HTTP 状态码
Rejected.LimitExceeded	The request was rejected because user create resource limit was exceeded.	400
AliasAlreadyExists	AliasName Already Exists.	400
InvalidKeyMaterial	key material is invalid.	400
InvalidImportToken	import token is invalid.	400
ExpiredImportToken	import token is expired.	400
Unsupported.Origin	This key origin is not valid for this api.	400
Unsupported.Alias	Alias is not valid for this api.	400
Unsupported.ProtectionLevel	This protection level is not valid for this region	400
Rejected.StateModifiedFailed	Keystate modified failed.	409
Rejected.Disabled	The request was rejected because the key state is Disabled.	409
Rejected.PendingDeletion	The request was rejected because the key state is PendingDeletion.	409
Rejected.PendingImport	The request was rejected because the key state is PendingImport.	409

2 用户主密钥的状态对API调用的影响

在KMS服务中，用户的每个主密钥都拥有启用（Enabled）、禁用（Disabled）、待删除（PendingDeletion）三个状态。

如果密钥是外部密钥（也叫用户自带密钥，#unique_15/unique_15_Connect_42_section_28952_03中Origin为EXTERNAL的），还可能处于待导入（PendingImport）状态。

通常情况下，新建的主密钥默认处于启用状态。当新建一个外部密钥时会处于等待导入状态。

只有处于启用状态的密钥才可以用于加密、解密操作。其它API根据密钥状态的不同，会有不同的返回结果。

处于待删除（PendingDeletion）状态的密钥，在预删除时间过后，会被永久删除。

密钥状态与API调用期望返回结果如下表所示。

期望结果	HttpStatusCode
Success	200
Rejected.Enabled	409
Rejected.Disabled	409
Rejected.PendingDeletion	409
Rejected.PendingImport	409
Rejected.StateModifiedFailed	409

普通API

API	启用（Enabled）	禁用（Disabled）	待删除（PendingDeletion）	待导入（PendingImport）
CreateKey	Success	Success	Success	Success
GenerateDataKey	Success	Rejected.Disabled	Rejected.PendingDeletion	Rejected.PendingImport
GenerateDataKeyWithoutPlaintext	Success	Rejected.Disabled	Rejected.PendingDeletion	Rejected.PendingImport

API	启用（Enabled）	禁用（Disabled）	待删除（PendingDeletion）	待导入（PendingImport）
Encrypt	Success	Rejected.Disabled	Rejected.PendingDeletion	Rejected.PendingImport
Decrypt	Success	Rejected.Disabled	Rejected.PendingDeletion	Rejected.PendingImport
ListKeys	Success	Success	Success	Success
DescribeKey	Success	Success	Success	Success
UpdateKeyDescription	Success	Success	Rejected.PendingDeletion	Success
EnableKey	Success	Success	Rejected.StateModifiedFailed	Rejected.StateModifiedFailed
DisableKey	Success	Success	Rejected.StateModifiedFailed	Rejected.StateModifiedFailed
ScheduleKeyDeletion	Success	Success	Rejected.StateModifiedFailed	Success
CancelKeyDeletion	Rejected.StateModifiedFailed	Rejected.StateModifiedFailed	Success	Rejected.StateModifiedFailed
CreateAlias	Success	Success	Rejected.StateModifiedFailed	Success
DeleteAlias	Success	Success	Success	Success
ListAliases	Success	Success	Success	Success
TagResource	Success	Success	Rejected.PendingDeletion	Success
UntagResource	Success	Success	Rejected.PendingDeletion	Success

API	启用（Enabled）	禁用（Disabled）	待删除（PendingDeletion）	待导入（PendingImport）
ListResourceTags	Success	Success	Success	Success
CreateKeyVersion	Success	Rejected.Disabled	Rejected.PendingDeletion	Rejected.PendingImport
DescribeKeyVersion	Success	Success	Success	Success
ListKeyVersions	Success	Success	Success	Success
UpdateRotationPolicy	Success	Rejected.Disabled	Rejected.PendingDeletion	Rejected.PendingImport

特殊API

`UpdateAlias`

- 只受到“目标密钥”的状态影响，与“原密钥”状态无关。
- 当“目标密钥”处于待删除状态时，返回Rejected.PendingDeletion，否则返回Success。

外部密钥专属API

API	启用（Enabled）	禁用（Disabled）	待删除（PendingDeletion）	待导入（PendingImport）
GetParametersForImport	Success	Success	Success	Success
ImportKeyMaterial	Success	Success	Rejected.StateModifiedFailed	Success
DeleteKeyMaterial	Success	Success	Success	Success

3 API 概览

本文列举了密钥管理服务（KMS）提供的 API 接口，具体 API 接口信息请参考相关文档。

阿里云也提供命令行工具，供您学习 API 或用于命令行自动化。关于命令行工具的安装和使用，详情请参考：阿里云 CLI。

密钥管理接口

密钥管理接口用于密钥的创建、属性修改以及生命周期管理。

API	描述
#unique_17	创建用户主密钥。用户可以选择由 KMS 生成密钥材料；也可以选择自己上传密钥材料（也就是BYOK，此时CreateKey是 BYOK 的第一步）。
GetParametersForImport	创建外部密钥（BYOK）的第二步：获取导入主密钥的材料。
ImportKeyMaterial	创建外部密钥（BYOK）的第三步：导入密钥材料到用户主密钥中，完成外部密钥的创建。
EnableKey	修改密钥的状态为：启用。
DisableKey	修改密钥的状态为：禁用。
ScheduleKeyDeletion	计划删除密钥。将密钥的状态设置为待删除状态，处于待删除状态的主密钥，会在计划的日期到期后删除。
CancelKeyDeletion	取消计划删除。处于待删除状态的密钥，在计划的日期到期之前，可以取消删除的计划，重新设置密钥状态为：启用。
DeleteKeyMaterial	直接删除用户主密钥的密钥材料。针对导入的外部密钥（BYOK），可以直接删除导入的密钥材料，删除密钥材料后的用户主密钥状态为：等待导入。
#unique_25	查询指定密钥的信息。
ListKeys	列出云帐号在本地域的所有密钥。

密码运算接口

密码运算接口用于对数据进行密码运算，例如：数据的加密和解密。

API	描述
Encrypt	使用指定用户主密钥加密数据，用于少量数据（不多于6KB）的在线加密。
GenerateDataKey	产生一个随机数，并用指定的用户主密钥加密后，返回随机数的密文以及明文。随机数可被用作数据密钥，在本地做大量数据加密或解密。
Decrypt	解密 Encrypt 或 GenerateDataKey接口产生的密文，不需要指定用于解密的用户主密钥。

别名管理接口

别名是独立的对象，但是必须和唯一的用户主密钥进行绑定，从而可以在特定 API 中代替 KeyId 参数来指代用户主密钥。

API	描述
CreateAlias	创建一个别名，并且将别名与一个用户主密钥绑定。
UpdateAlias	绑定指定别名到新的用户主密钥。
DeleteAlias	删除指定别名。
ListAliases	列出云帐号在本地域的所有别名。
ListAliasesByKeyId	列出与指定用户主密钥绑定的别名。

标签管理接口

用户主密钥支持标签。用户可以为用户主密钥添加多个标签，每一个标签为一组标签键（TagKey）和标签值（TagValue）。

API	描述
TagResource	为用户主密钥添加或修改标签。
UntagResource	删除用户主密钥的指定标签。
ListResourceTags	列出用户主密钥的所有标签。

4 API列表

4.1 CreateKey

创建一个主密钥。

主密钥可直接用于加密少量数据（少于 6 KB），但通常用于生成可以加密大量数据的 DataKey，详情请参见#unique_40 。

请求参数

名称	类型	是否必需	描述
Origin	String	否	密钥材料来源。 有效值： Aliyun_KMS 或 EXTERNAL。 说明： 有效值默认为 Aliyun_KMS。请注意区分大小写。 如果选择 EXTERNAL，您需要#unique_41。
Description	String	否	密钥的描述。长度必须在 0 到 8192 个字符之间。
KeyUsage	String	否	密钥的用途。默认值：ENCRYPT/DECRYPT。

ProtectionLevel	String	否	密钥的保护级别。 有效值：SOFTWARE 或 HSM。当指定值为 HSM 时： · 如果 Origin 为 Aliyun_KMS，则会在托管密码机中生成密钥，用于执行密码运算。 · 如果 Origin 为 EXTERNAL，您可以将外部密钥导入到托管密码机中，用于执行密码运算。 说明： 有效值默认为 SOFTWARE。请注意区分大小写。

返回参数

名称	类型	描述
KeyMetadata	KeyMetadata	密钥的 metadata。

KeyMetadata

名称	类型	描述
CreationDate	Timestamp	创建密钥时的日期和时间（UTC时间）。
Description	String	密钥的描述。
KeyId	String	密钥的全局唯一标识符。
KeyState	String	密钥的状态，详情请参见#unique_42 。
KeyUsage	String	密钥的用途，加密或解密。

DeleteDate	Timestamp	密钥预计被删除的时间（UTC时间）。 · 当该值为空时，表示密钥不会被删除。 · 只有当 KeyState 值为 PendingDeletion 时，会返回此参数。
Creator	String	密钥的创建者。
Arn	String	当前密钥的阿里云资源名称。
Origin	String	密钥材料来源。
MaterialExpireTime	String	密钥材料过期时间（UTC时间）。当该值为空时，表示密钥材料不会过期。
ProtectionLevel	String	密钥的保护级别。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=CreateKey
&Description=<your key description>
&KeyUsage=ENCRYPT/DECRYPT
&Origin=<key origin, default Aliyun_KMS>
&ProtectionLevel=HSM
&<公共请求参数>
```

返回示例

`JSON` 格式

```
//json response
{
        "KeyMetadata": {
                "CreationDate": "2016-03-25T10:42:40Z",
                "Description": "key description example",
                "KeyId": "08c33a6f-4e0a-4a1b-a3fa-7ddfa1d4****",
                "KeyState": "Enabled",
                "KeyUsage": "ENCRYPT/DECRYPT",
                "DeleteDate": "",
                "Creator":"123456",
                "Arn":"acs:kms:cn-hangzhou:123456:key/08c33a6f-4e0a-
4a1b-a3fa-7ddfa1d4****",
                "Origin":"Aliyun_KMS",
                "MaterialExpireTime":"",
                "ProtectionLevel":"HSM"
        },
        "RequestId": "3455b9b4-95c1-419d-b310-db6a53b09a39"
```

```
}
```

`XML` 格式

```
//xml response
<KMS>
 <KeyMetadata>
        <CreationDate>2016-03-25T10:40:47Z</CreationDate>
        <Description>key description example</Description>
        <KeyId>08c33a6f-4e0a-4a1b-a3fa-7ddfa1d4****</KeyId>
        <KeyState>Enabled</KeyState>
        <KeyUsage>ENCRYPT/DECRYPT</KeyUsage>
        <DeleteDate></DeleteDate>
        <Creator>123456</Creator>
        <Arn>acs:kms:cn-hangzhou:123456:key/08c33a6f-4e0a-4a1b-a3fa-
7ddfa1d4****</Arn>
        <Origin>Aliyun_KMS</Origin>
        <MaterialExpireTime></MaterialExpireTime>
        <ProtectionLevel>HSM</ProtectionLevel>
 </KeyMetadata>
 <RequestId>6cb4bf6b-d9c9-4660-af5f-2328378e7257</RequestId>
</KMS>
```

4.2 GetParametersForImport

获取导入主密钥（CMK）材料的参数，返回的参数可用于执行ImportKeyMaterial。

说明:

- 主密钥材料来源是必须外部，即`Origin`为`EXTERNAL`。
- API 会返回用于加密密钥材料的公钥 (public key)，导入密钥材料的令牌（token），以及令牌的过期时间。公钥是 base64 编码，令牌的有效期为 24 小时。
- 需要指定用于加密密钥材料的公钥类型（目前只支持RSA_2048），以及加密算法(目前支持RSAES_PKCS1_V1_5、RSAES_OAEP_SHA_1、RSAES_OAEP_SHA_256三种加密算法)。
- 本次调用返回的公钥和令牌，只能用于本次调用中指定的 CMK。
- 同次调用返回的公钥和令牌必须搭配使用。
- 加密密钥材料时所使用的算法必须是调用 API 时指定的加密算法。
- 每次调用返回的公钥与令牌都不相同。

请求参数

名称	类型	是否必需	描述
KeyId	String	是	CMK 全局唯一标识符。密钥材料来源必须是外部（`Origin`为`EXTERNAL`）。

WrappingAlgorithm	String	是	用于加密密钥材料的算法。
WrappingKeySpec	String	是	用于加密密钥材料的公钥类型。

返回参数

名称	类型	描述
KeyId	String	CMK 全局唯一标识符，后续调用 ImportKeyMaterial 时需要指定该参数。
ImportToken	String	后续调用 ImportKeyMaterial 的导入令牌。
PublicKey	String	密钥材料导入前，使用该公钥将其加密。
TokenExpireTime	String	导入令牌的过期时间。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=GetParametersForImport
&KeyId=<external key id>
&WrappingAlgorithm=<key material encryption algorithm>
&WrappingKeySpec=RSA_2048
&<公共请求参数>
```

返回示例

`JSON` 格式

```
//json response
{
        "ImportToken":"ImportToken",
        "PublicKey":"PublicKey",
        "KeyId":"KeyId",
        "TokenExpireTime":"2018-01-25T00:01:02Z",
        "RequestId":"8cdf51fd-bcd6-d79a-0ef4-e52c9b5466dc"
}
```

`XML` 格式

```
//xml response
<KMS>
  <ImportToken>ImportToken</ImportToken>
  <PublicKey>PublicKey</PublicKey>
  <KeyId>KeyId</KeyId>
  <TokenExpireTime>2018-01-25T00:01:02Z</TokenExpireTime>
  <RequestId>8cdf51fd-bcd6-d79a-0ef4-e52c9b5466dc</RequestId>
```

```
</KMS>
```

4.3 ImportKeyMaterial

调用#unique_10创建主密钥时，可以选择其密钥材料来源为外部，即将Origin设置为EXTERNAL，并在创建时不导入密钥材料。此 API 用于将密钥材料导入符合上述描述的 CMK 中。

- 要查看 CMK 的Origin，请参见#unique_43。
- 在导入密钥材料之前， 需要调用#unique_44先获得导入密钥材料需要的参数，即用于加密密钥材料的公钥（public key）和导入令牌（token）。

> 说明:
>
> - 密钥材料只能是 256 位的对称密钥。
> - 您可以为密钥材料设置过期时间，也可以设置其永不过期。
> - 您可以随时为指定的 CMK 重新导入密钥材料，并重新指定过期时间。但必须导入相同的密钥材料，某个指定的 CMK 不可以更换密钥材料。
> - 导入的密钥材料过期或者被删除后，指定的CMK将无法使用，需要再次导入相同的密钥材料才可正常使用。
> - 同样的密钥材料可导入不同的 CMK 中，但使用其中一个 CMK 加密的数据或 Datakey，无法使用另一个 CMK解密。

请求参数

名称	类型	是否必需	描述
EncryptedKeyMaterial	String	是	Base64 加密后的密钥材料。
ImportToken	String	是	通过调用GetParametersForImport获得的导入令牌。
KeyMaterialExpireUnix	Timestamp	否	密钥材料过期时间。- 不指定该参数，或取值为 0 表示密钥材料不会过期。- 取值不可早于调用该API 的时间（以服务器时间为准）。

返回参数

名称	类型	描述

RequestId	String	本次请求的 ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=ImportKeyMaterial
&EncryptedKeyMaterial=<your encrypted key material>
&ImportToken=<import token from GetParametersForImport>
&KeyMaterialExpireUnix=1518307200
&<公共请求参数>
```

返回示例

JSON格式

```
//json response
{
        "RequestId":"ec1017cf-ead4-f3ca-babc-c3b34f3dbecb"
}
```

XML格式

```
//xml response
<KMS>
  <RequestId>ec1017cf-ead4-f3ca-babc-c3b34f3dbecb</RequestId>
</KMS>
```

4.4 EnableKey

将一个指定的 CMK 标记为启用状态，可以使用它进行加解密。

请求参数

名称	类型	是否必需	描述
KeyId	string	是	CMK 的全局唯一标识符。

返回参数

名称	类型	描述
RequestId	String	本次 API 请求的 ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=EnableKey
&KeyId=<cmkid>
```

```
&<公共请求参数>
```

返回示例

JSON 格式

```
//json response
{
"RequestId": "efb1cbbd-a093-4278-bc03-639dd4fcc207"
}
```

XML 格式

```
//xml response

<KMS>
    <RequestId>efb1cbbd-a093-4278-bc03-639dd4fcc207</RequestId>
</KMS>
```

4.5 DisableKey

将一个指定的主密钥（CMK）标记为禁用状态。处于禁用状态的 CMK 无法用于加密、解密操作；恢复至启用状态之前，原来使用该 CMK 加密的密文也无法解密。

请求参数

名称	类型	是否必需	描述
KeyId	String	是	CMK 的全局唯一标识符。

返回参数

名称	类型	描述
RequestId	String	本次请求的 ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=DisableKey
&KeyId=<cmkid>
&<公共请求参数>
```

返回示例

JSON 格式

```
//json response
{
"RequestId": "2fe70ce2-3303-4fd6-b3ac-472fb2705c62"
```

```
}
```

XML 格式

```
//xml response
<KMS>
    <RequestId>2fe70ce2-3303-4fd6-b3ac-472fb2705c62</RequestId>
</KMS>
```

4.6 ScheduleKeyDeletion

申请删除一个指定的主密钥（CMK)。

- 在密钥预删除期间，密钥状态处于待删除，无法用于加密、解密、产生数据密钥操作。
- 主密钥一旦被删除无法恢复，使用该主密钥加密的内容与使用该主密钥产生的数据密钥（Datakey）均无法再被解密。因此，我们不提供直接删除主密钥的功能，而是采用申请删除的方式。并且我们建议您尽可能选择密钥禁用DisableKey。
- 在申请删除主密钥的同时，需要指定一个预删除周期，该周期最少为 7 天，最多为 30 天。从申请删除主密钥的时刻开始，到删除周期之前，可以通过CancelKeyDeletion撤销密钥删除的申请。
- 密钥会在到达预删除时间后的 24 小时之内被删除。API 服务器采用 UTC 时间。 例如用户 A 在2016年9月10日14:00申请了一个主密钥删除，预计删除时间在 7 天以后，那么KMS服务将在9月17日14:00之后的 24 小时内完成删除。

请求参数

名称	类型	是否必需	描述
KeyId	String	是	CMK 全局唯一标识符。
PendingWindowInDays	Integer	是	密钥预删除周期。在这段时间内，您可以撤销删除处于待删除状态的密钥；预删除时间过后无法撤销删除。有效值：最小值为 7，最大值为 30。

返回参数

名称	类型	描述
RequestId	String	本次请求的 ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=ScheduleKeyDeletion
&KeyId=<your-key-id>
&PendingWindowInDays=[7~30]
&<公共请求参数>
```

返回示例

`JSON` **格式**

```
//json response
{
"RequestId": "52ac67cb-3d3d-4ada-b4e2-7047660d3ce9"
}
```

`XML` **格式**

```
//xml response
<KMS>
    <RequestId>52ac67cb-3d3d-4ada-b4e2-7047660d3ce9</RequestId>
</KMS>
```

4.7 CancelKeyDeletion

撤销密钥删除。当密钥删除的申请撤销成功以后，密钥会处于启用状态。

请求参数

名称	类型	是否必需	描述
KeyId	String	是	CMK的全局唯一标示符。

返回参数

名称	类型	描述
RequestId	String	本次请求的ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=CancelKeyDeletion
&KeyId=<cmkid>
&<公共请求参数>
```

返回示例

JSON 格式

```
//json response
{
"RequestId": "3da5b8cc-8107-40ac-a170-793cd181d7b7"
}
```

XML 格式

```
//xml response
<KMS>
    <RequestId>3da5b8cc-8107-40ac-a170-793cd181d7b7</RequestId>
</KMS>
```

4.8 DeleteKeyMaterial

删除已导入的密钥材料。

- 此操作不会删除其对应的主密钥（CMK）。
- 如果 CMK 处于待删除状态，删除密钥材料不会改变密钥状态和预计删除时间；如果密钥不是处于待删除状态，删除密钥材料会使得密钥状态变更为等待导入。
- 删除密钥材料后，可以重新导入密钥材料，但必须与之前的密钥材料相同。

请求参数

名称	类型	是否必需	描述
KeyId	String	是	CMK 的全局唯一标识符。

返回参数

名称	类型	描述
RequestId	String	本次请求的 ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=DeleteKeyMaterial
&KeyId=<external key id>
&<公共请求参数>
```

返回示例

JSON 格式

```
//json response
{
        "RequestId": "4162a6af-bc99-40b3-a552-89dcc8aaf7c8"
```

```
}
```

XML 格式

```
//xml response
<KMS>
  <RequestId>4162a6af-bc99-40b3-a552-89dcc8aaf7c8</RequestId>
</KMS>
```

4.9 DescribeKey

返回指定主密钥（CMK）的相关信息。

请求参数

名称	类型	是否必需	描述
KeyId	String	是	CMK 的全局唯一标识符。该 API 支持使用别名，详情请参见#unique_46。

返回参数

名称	类型	描述
KeyMetadata	KeyMetadata	CMK 的 metadata。

KeyMetadata

名称	类型	描述
CreationDate	Timestamp	创建主密钥（CMK）的日期和时间（UTC）。
Description	String	CMK 的描述。
KeyId	String	CMK 全局唯一标识符。
KeyState	String	CMK 的状态，详情请参见#unique_42。
KeyUsage	String	CMK 的用途。
DeleteDate	Timestamp	CMK 的预计删除时间，详情请参见ScheduleKeyDeletion。只有当 KeyState 值为 PendingDeletion 时，返回该值。
Creator	String	CMK 创建者。
Arn	String	阿里云资源名称。

Origin	String	CMK 的密钥材料来源。
MaterialExpireTime	String	密钥材料的过期时间（UTC）。当该值为空时，表示密钥材料不会过期。
ProtectionLevel	String	密钥的保护级别。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=DescribeKey
&KeyId=<your-key-id>
&<公共请求参数>
```

返回示例

`JSON` 格式

```
//json response
{
        "KeyMetadata": {
                "CreationDate": "2016-03-25T10:42:40Z",
                "Description": "key description example",
                "KeyId": "08c33a6f-4e0a-4a1b-a3fa-7ddf****",
                "KeyState": "Enabled",
                "KeyUsage": "ENCRYPT/DECRYPT",
                "DeleteDate": "",
                "Creator":"123456",
                "Arn":"acs:kms:cn-hangzhou:123456:key/08c33a6f-4e0a-
4a1b-a3fa-7ddf****",
                "Origin":"Aliyun_KMS",
                "MaterialExpireTime":"",
                "ProtectionLevel":"HSM"
        },
        "RequestId": "3455b9b4-95c1-419d-b310-db6a53b09a39"
}
```

`XML` 格式

```
//xml response
<KMS>
 <KeyMetadata>
        <CreationDate>2016-03-25T10:40:47Z</CreationDate>
        <Description>key description example</Description>
        <KeyId>08c33a6f-4e0a-4a1b-a3fa-7ddf****</KeyId>
        <KeyState>Enabled</KeyState>
        <KeyUsage>ENCRYPT/DECRYPT</KeyUsage>
        <DeleteDate></DeleteDate>
        <Creator>123456</Creator>
        <Arn>acs:kms:cn-hangzhou:123456:key/08c33a6f-4e0a-4a1b-a3fa-
7ddf****</Arn>
        <Origin>Aliyun_KMS</Origin>
        <MaterialExpireTime></MaterialExpireTime>
        <ProtectionLevel>HSM</ProtectionLevel>
 </KeyMetadata>
 <RequestId>6cb4bf6b-d9c9-4660-af5f-2328378e7257</RequestId>
```

```
</KMS>
```

4.10 ListKeys

返回调用者在调用区域的所有的主密钥 ID。

请求参数

名称	类型	是否必需	描述
PageNumber	Integer	否	当前页数。有效值：大于 0， 默认值为 1。
PageSize	Integer	否	每页返回值的个数。有效值：大于 0，小于 101，默认为 10。

返回参数

名称	类型	描述
KeyId	String	主密钥的全局唯一标识符。
TotalCount	Integer	主密钥的总数。
PageNumber	Integer	当前页数。
PageSize	Integer	每页返回值的个数。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=ListKeys
&PageNumber=1
&PageSize=10
&<公共请求参数>
```

返回示例

`JSON` **格式**

```
//json response
{
        "Keys": {
                "Key": [
                        {
                                "KeyId": "08c33a6f-4e0a-4a1b-a3fa-
7ddfa1d4****"
                        },
                        {
                                "KeyId": "0e478b7a-4262-4802-b8cb-
00d3fb40****"
                        },
                        {
```

```
                                    "KeyId": "1abf9b4e-d3dd-4d4b-b9b2-
2829043a****"
                        }
                ]
        },
        "TotalCount": 3,
        "PageNumber": 1,
        "PageSize": 10,
        "RequestId": "1fbcd12a-1b7f-468f-84a3-1ff3444dfd8b"
}
```

XML 格式

```
//xml response
<KMS>
        <Keys>
                <Key>
                        <KeyId>08c33a6f-4e0a-4a1b-a3fa-7ddfa1d4****</
KeyId>
                </Key>
                <Key>
                        <KeyId>0e478b7a-4262-4802-b8cb-00d3fb40****</
KeyId>
                </Key>
                <Key>
                        <KeyId>1abf9b4e-d3dd-4d4b-b9b2-2829043a****</
KeyId>
                </Key>
        </Keys>
        <TotalCount>3</TotalCount>
        <PageNumber>1</PageNumber>
        <PageSize>10</PageSize>
        <RequestId>1050b8f1-b264-496d-a782-6299cbaf15f8</RequestId>
</KMS>
```

4.11 Encrypt

通过使用主密钥（CMK）将明文加密为密文。

- 可以加密最多为 6KB 任意数据，比如 RSA 密钥，数据库密码，或其他的敏感信息。
- 如果您是从一个 region 迁移加密数据到另一个 region，可以使用这个 API 在新的 region 中加密从另一个 region 中转移过来的明文 DataKey。新 region 里会生成一个加密后的 DataKey。你可以在新 region 将其#unique_47。

请求参数

名称	类型	是否必需	描述
KeyId	String	是	主密钥（CMK）的全局唯一标识符。该 API 支持使用别名，详情见#unique_46。

Plaintext	String	是	待加密明文（必须经过Base64 编码）。
EncryptionContext	String to string map	否	key/value对的JSON字符串，如果指定了该参数，则在调用Decrypt 时需要提供同样的参数，参见#unique_48。

返回参数

名称	类型	描述
KeyId	String	CMK 的全局唯一标识符。如果请求使用的别名，此处返回的是别名对应的主密钥 ID。
CiphertextBlob	String	加密过的密文。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=Encrypt
&KeyId=<cmkid or aliasname>
&Plaintext=<data need encrypt>
&EncryptionContext={"Example":"Example"}
&<公共请求参数>
```

返回示例

`JSON` 格式

```
//json response
{
 "KeyId": "your-key-id",
 "CiphertextBlob": "CiphertextBlob",
 "RequestId": "475f1620-b9d3-4d35-b5c6-3fbdd941423d"
}
```

`XML` 格式

```
//xml response
<KMS>
 <KeyId>your-key-id</KeyId>
 <CiphertextBlob>CiphertextBlob</CiphertextBlob>
 <RequestId>475f1620-b9d3-4d35-b5c6-3fbdd941423d</RequestId>
</KMS>
```

4.12 GenerateDataKey

生成一个密钥，你可以用这个密钥进行本地数据的加密。

这个 API 在Plaintext字段返回一个明文密钥，在CiphertextBlob字段返回一个加密后的密钥。

说明:

- 我们建议您使用以下方式在本地进行数据加密：
  - 调用本文 API GenerateDataKey，获得数据加密密钥。
  - 在返回结果中，使用Plaintext中的明文密钥对本地加密数据，然后删除内存中的明文密钥。
  - 在本地存储CiphertextBlob中返回的加密过的密钥，和加密后的数据。
- 在本地解密数据：
  - 调用#unique_47，将CiphertextBlob返回的加密过的密钥，解密为Plaintext密钥。
  - 用Plaintext密钥为本地数据解密，再删除本地存储中的Plaintext密钥。
- 当KeySpec和NumberOfBytes都不填写时，默认KeySpec为AES_256。
- 同时指定NumberOfBytes和KeySpec时，以NumberOfBytes为准。

请求参数

名称	类型	是否必需	描述
KeyId	String	是	主密钥（CMK）的全局唯一标识符。该 API 支持使用别名，详情见#unique_46。
KeySpec	String	否	产生数据密钥的长度与类型，AES_256表示 256 比特的对称密钥，AES_128表示 128 比特的对称密钥。
NumberOfBytes	Integer	否	产生数据密钥的长度，以字节为单位。有效值：1 到 1024。

EncryptionContext	String to string map	否	key/value对的json字符串，如果指定了该参数，则在调用 `Decrypt` 时需要提供同样的参数，参见#unique_48。

返回参数

名称	类型	描述
KeyId	String	CMK 的全局唯一标识符。如果请求使用的别名，此处返回的是别名对应的主密钥 ID。
Plaintext	String	明文 data key，该 data key 是经过 base64 编码的。
CiphertextBlob	String	加密过的 data key。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=GenerateDataKey
&KeyId=<cmkid or aliasname>
&KeySpec=AES_256
&EncryptionContext={"Example":"Example"}
&<公共请求参数>
```

返回示例

`JSON` **格式**

```
//json response
{
        "CiphertextBlob": "CiphertextBlob",
        "KeyId": "599fa825-17de-417e-9554-bb032cc6****",
        "Plaintext": "Base64 encoded plaintext",
        "RequestId": "7021b6ec-4be7-4d3c-8a68-1e85d4d515a0"
}
```

`XML` **格式**

```
//xml response
<KMS>
        <CiphertextBlob>CiphertextBlob</CiphertextBlob>
        <KeyId>599fa825-17de-417e-9554-bb032cc6****</KeyId>
        <Plaintext>Base64 encoded plaintext</Plaintext>
        <RequestId>7021b6ec-4be7-4d3c-8a68-1e85d4d515a0</RequestId>
</KMS>
```

4.13 Decrypt

解密CiphertextBlob中的密文。

密文可以是以下 API 生成的：

- #unique_40
- Encrypt

请求参数

名称	类型	是否必需	描述
CiphertextBlob	String	是	待解密的密文。
EncryptionContext	String	否	key/value 对的JSON字符串，如果在Encrypt或者GenerateDataKey API 中指定了该参数，则需要提供同样的参数才能解密，参见#unique_48。

返回参数

名称	类型	描述
KeyId	String	密钥的全局唯一标识符。加密密文使用的主密钥 CMK ID。
Plaintext	String	解密后的明文。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=Decrypt
&CiphertextBlob=<your ciphertextblob>
&EncryptionContext={"Example":"Example"}
&<公共请求参数>
```

返回示例

JSON 格式

```
//json response
{
"KeyId": "202b9877-5a25-46e3-a763-e20791b5****"
```

```
"Plaintext": "Plaintext"
"RequestId": "207596a2-36d3-4840-b1bd-f87044699bd7"
}
```

XML 格式

```
//xml response
<KMS>
        <KeyId>202b9877-5a25-46e3-a763-e20791b5****</KeyId>
        <Plaintext>Plaintext</Plaintext>
        <RequestId>4bd560a1-729e-45f1-a3d9-b2a33d61046b</RequestId>
</KMS>
```

4.14 CreateAlias

给主密钥（CMK）创建一个别名。

说明:

- 每个别名只能表示一个 CMK。
- 在同一用户的一个地区内，别名不可重复。
- 可以使用 #unique_49 更新别名和主密钥的映射关系。

请求参数

名称	类型	是否必需	描述
AliasName	String	是	- CMK 的别名，可以使用别名调用 Encrypt、GenerateDataKey、DescribeKey。- 前缀以外的字符长度：最小长度为 1 字符，最大长度为255 字符。 - 必须包含前缀 alias/。
KeyId	String	是	key的全局唯一标识符。

返回参数

名称	类型	描述
RequestId	String	本次请求的 ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=CreareAlias
&KeyId=<cmkid>
&AliasName=<alias/example>
&<公共请求参数>
```

返回示例

`JSON` **格式**

```
//json response
{
    "RequestId": "53790170-1096-4ed2-9c3a-244d75c8740a"
}
```

`XML` **格式**

```
//xml response
<KMS>
    <RequestId>53790170-1096-4ed2-9c3a-244d75c8740a</RequestId>
</KMS>
```

4.15 UpdateAlias

更新已存在的别名所代表的主密钥（CMK）。

请求参数

名称	类型	是否必需	描述
AliasName	String	是	要操作的别名。 1. 必须包含前缀 `alias/`。 2. 不包含前缀的字符长度为 [1, 255]。
KeyId	String	是	CMK 的全局唯一标识符。

返回参数

名称	类型	描述
RequestId	String	本次请求的 ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=UpdateAlias
&AliasName=<alias name>
&KeyId=<target keyid>
&<公共请求参数>
```

返回示例

`JSON` 格式

```
//json response
{
        "RequestId": "1d2baaf3-d357-46c2-832e-13560c2bd9cd"
}
```

`XML` 格式

```
//xml response
<KMS>
        <RequestId>1d2baaf3-d357-46c2-832e-13560c2bd9cd</RequestId>
</KMS>
```

4.16 DeleteAlias

删除别名。

请求参数

名称	类型	是否必需	描述
AliasName	String	是	要操作的别名。 · 必须包含前缀 `alias/`。 · 不包含前缀的字符长度为 [1, 255]。

返回参数

名称	类型	描述
RequestId	String	本次请求的ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=DeleteAlias
&AliasName=<alias name>
```

```
&<公共请求参数>
```

返回示例

JSON 格式

```
//json response
{
        "RequestId": "4c8ae23f-3a42-6791-a      4ba-1faa77831c28"
}
```

XML 格式

```
//xml response
<KMS>
        <RequestId>4c8ae23f-3a42-6791-a 4ba-1faa77831c28</RequestId>
</KMS>
```

4.17 ListAliases

返回当前用户在当前区域的所有别名。

请求参数

名称	类型	是否必需	描述
PageNumber	Integer	否	当前页数。 参数值为大于 0 的整数，默认为 1。
PageSize	Integer	否	每页返回的结果个数。参数值为 0 到 101 之间的整数。默认为 10。

返回参数

名称	类型	描述
AliasName	String	别名的唯一标识符。
AliasArn	String	别名的 ARN。
KeyId	String	别名对应的 CMK。
TotalCount	Integer	返回的 CMK 总数。
PageNumber	Integer	当前页数。
PageSize	Integer	每页的返回结果个数。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=ListAliases
&PageNumber=1
&PageSize=10
&<公共请求参数>
```

返回示例

JSON 格式

```
//json response
{
        "Aliases": {
                "Alias": [
                        {
                                "AliasName": "alias/ExampleAlias1",
                                "KeyId": "08c33a6f-4e0a-4a1b-a3fa-
7ddfa1d****",
                                "AliasArn": "acs:kms:cn-hangzhou:
123456:alias/ExampleAlias1"
                        }
        ]
        },
        "TotalCount": 1,
        "PageNumber": 1,
        "PageSize": 10,
        "RequestId": "1b57992c-834b-4811-a889-f8bac1ba0353"
}
```

XML 格式

```
//xml response
<KMS>
        <Aliases>
                <Alias>
                        <AliasName>alias/ExampleAlias1</AliasName>
                        <KeyId>08c33a6f-4e0a-4a1b-a3fa-7ddfa1****</
KeyId>
                        <AliasArn>acs:kms:cn-hangzhou:123456:alias/
ExampleAlias1</AliasArn>
                </Alias>
        </Aliases>
        <TotalCount>1</TotalCount>
        <PageNumber>1</PageNumber>
        <PageSize>10</PageSize>
        <RequestId>1b57992c-834b-4811-a889-f8bac1ba0353</RequestId>
</KMS>
```

4.18 ListAliasesByKeyId

列出与指定主密钥（CMK）对应的所有别名。

请求参数

名称	类型	是否必需	描述
PageNumber	Integer	否	当前页数。 参数值为大于 0 的整数，默认为 1。
PageSize	Integer	否	每页返回的结果个数。参数值为 0 到 101 之间的整数。默认为 10 。
KeyId	String	是	CMK 的全局唯一标识符。

返回参数

名称	类型	描述
AliasName	String	别名的唯一标识符。
AliasArn	String	别名的 ARN。
KeyId	String	别名对应的 CMK。
TotalCount	Integer	返回的 CMK 总数。
PageNumber	Integer	当前页数。
PageSize	Integer	每页的返回结果个数。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=ListAliasesByKeyId
&PageNumber=1
&PageSize=10
&KeyId=<cmkid>
&<公共请求参数>
```

返回示例

`JSON` 格式

```
//json response
{
```

```
        "Aliases": {
                "Alias": [
                        {
                                "AliasName": "alias/ExampleAlias1",
                                "KeyId": "08c33a6f-4e0a-4a1b-a3fa-
7ddfa1d4****",
                                "AliasArn": "acs:kms:cn-hangzhou:
123456:alias/ExampleAlias1"
                        }
        ]
        },
        "TotalCount": 1,
        "PageNumber": 1,
        "PageSize": 10,
        "RequestId": "1b57992c-834b-4811-a889-f8bac1ba0353"
}
```

XML 格式

```
//xml response
<KMS>
        <Aliases>
                <Alias>
                        <AliasName>alias/ExampleAlias1</AliasName>
                        <KeyId>08c33a6f-4e0a-4a1b-a3fa-7ddfa1d4****</
KeyId>
                        <AliasArn>acs:kms:cn-hangzhou:123456:alias/
ExampleAlias1</AliasArn>
                </Alias>
        </Aliases>
        <TotalCount>1</TotalCount>
        <PageNumber>1</PageNumber>
        <PageSize>10</PageSize>
        <RequestId>1b57992c-834b-4811-a889-f8bac1ba0353</RequestId>
</KMS>
```

4.19 TagResource

为用户主密钥添加或修改标签。

描述

每个用户主密钥可以有多个标签，一个标签由一组标签键和标签值进行定义。

> **说明:**
> 每个用户主密钥最多可以添加 10 个标签。

关于标签键和标签值，详情请参考：Tag 对象说明。

请求格式

```
KeyId="string"&Tags=[{ "TagKey": "string","TagValue": "string"} ]
```

请求参数

名称	类型	是否必选	示例值	描述
KeyId	String	是	external key id	全局唯一标识符。
Tags	JSON	是	[{"TagKey": "Project", "TagValue": "Test"}]	一个或者多个标签。格式为：Tag 对象数组，其中 Tag 对象有下列属性： · TagKey：标签的标签键。 · TagValue：标签的标签值。 关于标签键和标签值，详情请参考：Tag 对象说明。

Tag 对象说明

名称	类型	是否必选	示例值	描述
TagKey	String	是	Project	标签的标签键。 长度范围： 1~128 字符。 字符限制：a-zA-Z0-9/_-.+=@:
TagValue	String	是	Test	标签的标签值。 长度范围： 0~256 字符。 字符限制：a-zA-Z0-9/_-.+=@: 说明: 每个标签的标签键互不相同。调用 TagResource 接口时，如果输入的标签键在指定密钥上已经存在，则使用输入的标签值覆盖原标签值。

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=TagResource
&KeyId=<external key id>
```

```
&Tags=<tags>
&<其他公共参数>
```

返回参数

名称	类型	示例值	描述	
RequestId	String	4162a6af-bc99-40b3-a552-89dcc8aaf7c8	请求 ID。	

返回示例

JSON 格式

```
{
    "RequestId": "4162a6af-bc99-40b3-a552-89dcc8aaf7c8"
}
```

XML 格式

```
<KMS>
  <RequestId>4162a6af-bc99-40b3-a552-89dcc8aaf7c8</RequestId>
</KMS>
```

4.20 UntagResource

删除用户主密钥的指定标签。

请求格式

```
KeyId="string"&TagKeys=["tagkey1","tagkey2"]
```

请求参数

名称	类型	是否必选	示例值	描述
KeyId	String	是	external key id	全局唯一标识符。
TagsKeys	JSON	是	["tagkey1","tagkey2"]	一个或者多个标签键，只需要指定标签键，不需要指定标签值。格式为：String 数组。 数组中的 String 长度范围为：1~128 字符。

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=UntagResource
&KeyId=<external key id>
&TagKeys=<tagkeys>
```

```
&<其他公共参数>
```

返回参数

名称	类型	示例值	描述	
RequestId	String	4162a6af-bc99-40b3-a552-89dcc8aaf7c8	请求 ID。	

返回示例

JSON 格式

```
{
    "RequestId": "4162a6af-bc99-40b3-a552-89dcc8aaf7c8"
}
```

XML 格式

```
<KMS>
  <RequestId>4162a6af-bc99-40b3-a552-89dcc8aaf7c8</RequestId>
</KMS>
```

4.21 ListResourceTags

列出用户主密钥的标签。

请求格式

```
KeyId="string"
```

请求参数

名称	类型	是否必选	示例值	描述
KeyId	String	是	key id	全局唯一标识符。

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=ListResourceTags
&KeyId=<key id>
```

```
&<其他公共参数>
```

返回参数

名称	类型	示例值	描述
RequestId	String	4162a6af-bc99-40b3-a552-89dcc8aaf7c8	请求 ID。
TagKey	String	Project	标签键。
TagValue	String	Test	标签值。

返回示例

JSON 格式

```
{
    "RequestId": "4162a6af-bc99-40b3-a552-89dcc8aaf7c8",
    "Tags": {
            "Tag": [
                {
                    "KeyId": "33caea95-c3e5-4b3e-a9c6-cec76e4eaf83",
                    "TagKey": "Project",
                    "TagValue": "Test"
                }
            ]
        }
}
```

XML 格式

```
<KMS>
        <RequestId>0f900dad-c747-4170-9962-1bfb6b31436b</RequestId>
        <Tags>
            <Tag>
                <KeyId>33caea95-c3e5-4b3e-a9c6-cec76e4eaf83</KeyId>
                <TagKey>Project</TagKey>
                <TagValue>Test</TagValue>
            </Tag>
        </Tags>
</KMS>
```

4.22 DescribeRegions

查询当前账户的可用地域列表。

请求参数

名称	类型	是否必需	描述

Action	String	是	操作接口名。取值：DescribeRegions。

返回参数

名称	类型	描述
RegionId	String	可用区 ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=DescribeRegions
&<公共请求参数>
```

返回示例

`JSON` 格式

```
//json response
{
        "Regions": {
                "Region": [
                        {
                                "RegionId": "cn-beijing"
                        },
                        {
                                "RegionId": "cn-hangzhou"
                        }
                ]
        },
        "RequestId": "815240e2-aa37-4c26-9cca-05d4df3e8fe6"
}
```

`XML` 格式

```
//xml response
<KMS>
 <Regions>
  <Region>
   <RegionId>cn-beijing</RegionId>
  </Region>
  <Region>
   <RegionId>cn-hangzhou</RegionId>
  </Region>
 </Regions>
 <RequestId>815240e2-aa37-4c26-9cca-05d4df3e8fe6</RequestId>
</KMS>
```