ALIBABA CLOUD

阿里云

密钥管理服务

API参考

文档版本：20220208

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概览

本文列出了密钥管理服务KMS（Key Management Service）提供的API接口及相关描述。

密钥服务接口

- **用户主密钥管理**

  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
CreateKey	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（即是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。
SetDeletionProtection	为用户主密钥（CMK）开启或关闭删除保护。
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。
DescribeKey	查询指定密钥的信息。
ListKeys	列出云账号在本地域的所有密钥。
UpdateKeyDescription	更新用户主密钥的描述信息。

- **密钥版本管理**

  密钥版本管理接口用于对主密钥进行密钥轮转。

API	描述
DescribeKeyVersion	查看一个密钥版本。
ListKeyVersions	列出主密钥的所有密钥版本。
UpdateRotationPolicy	更新对称主密钥的轮转策略。如果配置自动轮转，KMS将周期性自动生成新的密钥版本。
CreateKeyVersion	创建新的密钥版本，适用于非对称密钥。

- **密码运算**

  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。
GenerateDataKeyWithoutPlaintext	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。随机数可被用作数据密钥，在本地做大量数据加密或解密。
ExportDataKey	使用传入的公钥加密导出数据密钥。
GenerateAndExportDataKey	随机生成一个数据密钥，通过您指定的主密钥（CMK）和公钥加密后，返回CMK加密数据密钥的密文和公钥加密数据密钥的密文。
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。
ReEncrypt	对密文进行转加密。即先解密密文，然后将解密得到的数据或者数据密钥使用新的主密钥再次进行加密，返回加密结果。
AsymmetricSign	非对称密钥的私钥运算：产生数字签名。
AsymmetricVerify	非对称密钥的公钥运算：验证私钥产生的数字签名。
AsymmetricDecrypt	非对称密钥的私钥运算：解密公钥加密的数据。
AsymmetricEncrypt	非对称密钥的公钥运算：加密数据。
GetPublicKey	获取非对称密钥的公钥，可用于离线验证数字签名，或者加密数据。

- **别名管理**

  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	更新已存在的别名所代表的主密钥（CMK）ID。
DeleteAlias	删除别名。
ListAliases	列出云账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。

凭据管家接口

KMS凭据管家提供凭据的托管、保护、分发和轮转能力。

API	描述
CreateSecret	创建凭据，并存入凭据的初始版本。
ListSecrets	查询当前用户云账号所在地域的所有凭据。
DeleteSecret	删除凭据对象。
DescribeSecret	获取凭据的元数据信息。
GetSecretValue	获取凭据值。
PutSecretValue	为凭据存入一个新版本的凭据值。
UpdateSecret	更新凭据的元数据。
UpdateSecretVersionStage	更新凭据的版本状态。
RestoreSecret	恢复被删除的凭据。
ListSecretVersionIds	查询凭据的所有版本信息。
GetRandomPassword	获得一个随机密码串。

证书接口

证书接口用于证书的创建、删除、更新、查询、签名验签等操作。

API	描述
CreateCertificate	创建证书。
UploadCertificate	将CA机构颁发的证书和证书链导入证书管家。
GetCertificate	查询证书管家托管的证书。
DescribeCertificate	查询证书信息。
UpdateCertificateStatus	更新证书状态。
DeleteCertificate	删除证书及其对应的私钥和证书链。
CertificatePrivateKeySign	使用指定证书生成数字签名。
CertificatePublicKeyVerify	使用指定证书验证数字签名。
CertificatePublicKeyEncrypt	使用指定证书加密数据。
CertificatePrivateKeyDecrypt	使用指定证书解密数据。

标签管理接口

用户主密钥支持标签。您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	为用户主密钥或者凭据添加或修改标签。
UntagResource	删除用户主密钥或者凭据的指定标签。
ListResourceTags	列出用户主密钥或者凭据的所有标签。

其他

API	描述
DescribeRegions	查询当前账户的可用地域列表。
OpenKmsService	为当前阿里云账号开通密钥管理服务。

API	描述
DescribeAccountKmsStatus	查询当前阿里云账号的密钥管理服务状态。

2.用户主密钥的状态对API调用的影响

在密钥管理服务（KMS）中，您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（用户自带密钥，即DescribeKey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
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
ListResourceTags	Success	Success	Success	Success
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

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

3.调用方式

密钥管理服务KMS接口调用是向KMS API的服务端地址发送HTTPS GET或POST请求。您需要按照API说明在请求中加入相应请求参数，调用后系统会返回处理结果。请求及返回结果都使用UTF-8字符集进行编码。

请求结构

KMS的API是RPC风格，您可以通过发送HTTPS GET或POST请求调用KMS API。

其请求结构如下：

```
https://Endpoint/?Action=xx&Parameters
```

请求结构中参数说明如下表所示。

参数	描述
Endpoint	KMS API的服务接入地址。更多信息，请参见服务接入地址。
Action	要执行的操作，例如：调用CreateKey创建一个主密钥。
Version	要使用的API版本。KMS的API版本是*2016-01-20*。
Parameters	请求参数，每个参数之间用and（&）分隔。 请求参数由公共请求参数和API自定义参数组成。公共参数中包含API版本号、身份验证等信息。更多信息，请参见公共参数。

请求示例

调用CreateKey创建一个主密钥的示例如下。

> **说明** 为了便于您查看，本文档中的示例都做了格式化处理。

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

服务接入地址

地域	RegionId	公网接入地址	VPC接入地址
日本（东京）	ap-northeast-1	kms.ap-northeast-1.aliyuncs.com	kms-vpc.ap-northeast-1.aliyuncs.com
新加坡	ap-southeast-1	kms.ap-southeast-1.aliyuncs.com	kms-vpc.ap-southeast-1.aliyuncs.com
澳大利亚（悉尼）	ap-southeast-2	kms.ap-southeast-2.aliyuncs.com	kms-vpc.ap-southeast-2.aliyuncs.com
马来西亚（吉隆坡）	ap-southeast-3	kms.ap-southeast-3.aliyuncs.com	kms-vpc.ap-southeast-3.aliyuncs.com
印度尼西亚（雅加达）	ap-southeast-5	kms.ap-southeast-5.aliyuncs.com	kms-vpc.ap-southeast-5.aliyuncs.com
菲律宾（马尼拉）	ap-southeast-6	kms.ap-southeast-6.aliyuncs.com	kms-vpc.ap-southeast-6.aliyuncs.com
印度（孟买）	ap-south-1	kms.ap-south-1.aliyuncs.com	kms-vpc.ap-south-1.aliyuncs.com
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
华北6（乌兰察布 ）	cn-wulanchabu	kms.cn-wulanchabu.aliyuncs.com	kms-vpc.cn-wulanchabu.aliyuncs.com
华南1（深圳）	cn-shenzhen	kms.cn-shenzhen.aliyuncs.com	kms-vpc.cn-shenzhen.aliyuncs.com
华南2（河源）	cn-heyuan	kms.cn-heyuan.aliyuncs.com	kms-vpc.cn-heyuan.aliyuncs.com
华南3（广州）	cn-guangzhou	kms.cn-guangzhou.aliyuncs.com	kms-vpc.cn-guangzhou.aliyuncs.com
西南1（成都）	cn-chengdu	kms.cn-chengdu.aliyuncs.com	kms-vpc.cn-chengdu.aliyuncs.com
德国（法兰克福）	eu-central-1	kms.eu-central-1.aliyuncs.com	kms-vpc.eu-central-1.aliyuncs.com

地域	RegionId	公网接入地址	VPC接入地址
英国（伦敦）	eu-west-1	kms.eu-west-1.aliyuncs.com	kms-vpc.eu-west-1.aliyuncs.com
阿联酋（迪拜）	me-east-1	kms.me-east-1.aliyuncs.com	kms-vpc.me-east-1.aliyuncs.com
中国（香港）	cn-hongkong	kms.cn-hongkong.aliyuncs.com	kms-vpc.cn-hongkong.aliyuncs.com
美国（弗吉尼亚）	us-east-1	kms.us-east-1.aliyuncs.com	kms-vpc.us-east-1.aliyuncs.com
美国（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
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

通信协议

KMS服务只支持使用HTTPS通道发送请求。

KMS仅支持TLS 1.0及以上版本，不支持SSL v2和SSL v3。

4.签名机制

为保证API的安全调用，在调用API时阿里云会对每个API请求通过签名（Signature）进行身份验证。当您使用HTTPS协议提交请求时，需要在请求中包含签名信息。

概述

RPC API要按以下格式在API请求的Query中增加签名（Signature）。

```
https://Endpoint/?SignatureVersion=1.0&SignatureMethod=HMAC-SHA1&Signature=CT9X0VtwR86fNWSnsc6v8YGOjuE%3D&
```

其中：

- SignatureMethod：签名方式，目前支持HMAC-SHA1。
- SignatureVersion：签名算法版本，目前版本是1.0。
- Signature：使用AccessKey Secret对请求进行对称加密后生成的签名。

签名算法遵循RFC 2104 HMAC-SHA1规范，使用AccessKey Secret对编码、排序后的整个请求串计算HMAC值作为签名。签名的元素是请求自身的一些参数，由于每个API请求内容不同，所以签名的结果也不尽相同。可参考本文的操作步骤，计算签名值。

```
Signature = Base64( HMAC-SHA1( AccessKey Secret, UTF-8-Encoding-Of(StringToSign)) )
```

步骤一：构造待签名字符串

1. 使用请求参数构造规范化的请求字符串（Canonicalized Query String）。
   i. 按照参数名称的字典顺序对请求中所有的请求参数（包括公共请求参数和接口的自定义参数，但不包括公共请求参数中的Signature参数）进行排序。

   > **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。

   ii. 对排序之后的请求参数的名称和值分别用UTF-8字符集进行URL编码。编码规则请参考下表。

字符	编码方式
A~Z、a~z和0~9以及“-”、“_”、“.”和“~”	不编码。
其它字符	编码成 `%XY` 的格式，其中 `XY` 是字符对应ASCII码的16进制表示。例如英文的双引号（"）对应的编码为 `%22`。
扩展的UTF-8字符	编码成 `%XY%ZA…` 的格式。

英文空格	编码成 `%20`，而不是加号（+）。
	该编码方式和一般采用的 `application/x-www-form-urlencoded` MIME格式编码算法（例如Java标准库中的 `java.net.URLEncoder` 的实现）存在区别。编码时可以先用标准库的方式进行编码，然后把编码后的字符串中的加号（+）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。本算法可以用下面的 `percentEncode` 方法来实现：
	`private static final String ENCODING = "UTF-8";`
	`private static String percentEncode(String value) throws UnsupportedEncodingException`
	`{`
	`return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("*", "%2A").replace("%7E", "~") : null;`
	`}`

   iii. 将编码后的参数名称和值用英文等号（=）进行连接。
   iv. 将等号连接得到的参数组合按步骤i排好的顺序依次使用“&”符号连接，即得到规范化请求字符串。
2. 将构造的规范化字符串按照下面的规则构造成待签名的字符串。

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

其中：

- HTTPMethod 是提交请求用的HTTP方法，例如GET。
- percentEncode(“/”) 是按照URL编码规则对字符“/”进行编码得到的值，即%2F。
- percentEncode(CanonicalizedQueryString) 是对构造的规范化请求字符串按URL编码规则编码后得到的字符串。

步骤二：计算签名值

1. 按照RFC2104的定义，计算待签名字符串（StringToSign）的HMAC值。

   > **说明** 计算签名时使用的Key就是您持有的AccessKey Secret并加上一个“&”字符（ASCII:38），使用的哈希算法是SHA1。

2. 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）。
3. 将得到的签名值作为Signature参数添加到请求参数中。

   > **说明** 得到的签名值在作为最后的请求参数值提交时要和其它参数一样，按照RFC3986的规则进行URL编码。

示例

下文以CreateKey为例，介绍签名的一个具体示例及结果。

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
```

对应的 `StringToSign` 为：

```
GET&%2F&AccessKeyId%3Dtestid%26Action%3DCreateKey%26Format%3Djson%26SignatureMethod%3DHMAC-SHA1%26SignatureVersion%3D1.0%26Timestamp%3D2016-03-28T03%
253A13%253A08Z%26Version%3D2016-01-20
```

例如：AccessKey ID为：testid，AccessKey Secret为：testsecret，则用于计算HMAC的key为：testsecret&。

计算得到的签名值为： `41wk2SSX1GJh7fwnc5eqOfiJPF****` 。

签名后的请求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
&Signature=41wk2SSX1GJh7fwnc5eqOfiJPF****
```

5.公共参数

公共参数分为公共请求参数和公共返回参数。

公共请求参数

名称	类型	是否必须	描述
Format	String	否	返回消息的格式。取值： • JSON（默认值） • XML
Version	String	是	API版本号，使用YYYY-MM-DD日期格式。取值：2016-01-20。
AccessKeyId	String	是	访问密钥ID。
Signature	String	是	消息签名。
SignatureMethod	String	是	签名方式。取值：HMAC-SHA1。
Timestamp	String	是	请求的时间戳，为日期格式。使用UTC时间并按照ISO8601标准，格式为：YYYY-MM-DDThh:mm:ssZ。 例如：北京时间2013年01月10日20点00分00秒，表示为2013-01-10T12:00:00Z。
SignatureVersion	String	是	签名算法版本。取值：1.0。

公共请求参数示例如下：

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

公共返回参数

API返回结果采用统一格式，调用成功返回的数据格式有XML和JSON两种，可以在发送请求时指定返回的数据格式，默认为JSON格式。每次接口调用，无论成功与否，系统都会返回一个唯一识别码RequestId。

- 返回 `2xx` HTTP状态码表示调用成功。
- 返回 `4xx` 或 `5xx` HTTP状态码表示调用失败。

公共返回参数示例如下：

- XML格式

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

- JSON格式

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

6.返回结果

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，表示调用成功。如果返回的状态码为4xx或5xx，表示调用失败。本文档中的返回示例为了便于您查看，做了格式化处理，实际返回结果没有进行换行、缩进等处理。

正常返回示例

`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 */
}
```

异常返回示例

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API报错后，将在response中返回HTTP状态码、错误码和错误信息，还会包括该次请求在全局的唯一标识RequestId。若您根据错误码和错误信息无法明确问题，可将RequestId提供给技术支持人员，帮忙快速定位到该条请求日志。

`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
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

错误代码	描述	HTTP状态码
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
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

7.获取AccessKey

您可以为阿里云账号（主账号）和RAM用户创建一个访问密钥（AccessKey）。在调用阿里云API时您需要使用AccessKey完成身份验证。

背景信息

AccessKey包括AccessKey ID和AccessKey Secret。

- AccessKey ID：用于标识用户。
- AccessKey Secret：用于验证用户的密钥。AccessKey Secret必须保密。

> **警告** 阿里云账号AccessKey泄露会威胁您所有资源的安全。建议您使用RAM用户AccessKey进行操作，可以有效降低AccessKey泄露的风险。

操作步骤

1. 使用阿里云账号登录控制台。
2. 将鼠标置于页面右上方的账号图标，单击**AccessKey管理**。
3. 在**安全提示**对话框，选择使用阿里云账号AccessKey或RAM用户AccessKey。

   - 使用阿里云账号AccessKey
     a. 单击**继续使用AccessKey**。
     b. 在**AccessKey**页面，单击**创建AccessKey**。
     c. 获取验证码，单击**确定**。
     d. 在**查看Secret**对话框，查看AccessKey ID和AccessKey Secret。可以单击**下载CSV文件**，下载AccessKey信息。或者单击**复制**，复制AccessKey信息。

   - 使用RAM用户AccessKey
     a. 单击**开始使用子用户AccessKey**。
     b. 系统自动跳转到RAM控制台的**用户**页面，找到需要获取AccessKey的RAM用户。

     > **说明** 如果没有RAM用户，请先创建RAM用户，详情请参见创建RAM用户。

     c. 单击用户登录名称。
     d. 在**认证管理**页签下的**用户AccessKey**区域，单击**创建AccessKey**。
     e. 获取验证码，单击**确定**。
     f. 在**查看Secret**页面，查看AccessKey ID和AccessKey Secret。可以单击**下载CSV文件**，下载AccessKey信息。或者单击**复制**，复制AccessKey信息。

     > **说明**
     > - RAM用户的AccessKey Secret只在创建时显示，不提供查询，请妥善保管。
     > - 若AccessKey泄露或丢失，则需要创建新的AccessKey，最多允许为每个RAM用户创建2个AccessKey。

8.密钥

8.1. CreateKey

调用CreateKey接口创建一个主密钥。

使用说明

主密钥可以是对称密钥或非对称密钥。对称密钥主要用于生成可以加密大量数据的DataKey，有时也直接用于加密少量数据（少于6KB）。更多信息，请参见GenerateDataKey。非对称密钥用于加密解密或签名验签，但无法生成数据密钥。

各种密钥类型支持的操作如下表所示：

密钥类型	密钥规格	说明	加密解密	签名验签
对称密钥	Aliyun_AES_256	AES密钥，长度为256比特	支持	不支持
对称密钥	Aliyun_AES_128	AES密钥，长度为128比特。仅专属KMS支持该密钥规格。	支持	不支持
对称密钥	Aliyun_AES_192	AES密钥，长度为192比特。仅专属KMS支持该密钥规格。	支持	不支持
对称密钥	Aliyun_SM4	SM4密钥	支持	不支持
非对称密钥	RSA_2048	RSA密钥，模长为2048比特	支持	支持
非对称密钥	RSA_3072	RSA密钥，模长为3072比特	支持	支持
非对称密钥	EC_P256	NIST推荐椭圆曲线P-256（secp256r1）	不支持	支持
非对称密钥	EC_P256K	SECG椭圆曲线secp256k1	不支持	支持
非对称密钥	EC_SM2	GBT 32918定义的素数域256位椭圆曲线	支持	支持

说明

- 对称密钥KeySpec在标准密钥类型前加上 `Aliyun_` 前缀，表示使用标准密钥的密码算法，但是会生成非标准密文；非对称密钥产生标准密文或者签名。
- RSA密钥使用方式仅支持ENCRYPT/DECRYPT、SIGN/VERIFY两者之一，单个密钥无法同时支持两种操作。
- SM4和SM2为中国国家密码管理局批准的密码算法，KMS通过部署在中国内地的托管密码机提供支持。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateKey	要执行的操作，取值：CreateKey。
Description	String	否	key description example	密钥的描述。 长度为0~8192个字符。

名称	类型	是否必选	示例值	描述
KeySpec	String	否	Aliyun_AES_256	密钥规格，取值： • Aliyun_AES_256 • Aliyun_AES_128 • Aliyun_AES_192 • Aliyun_SM4 • RSA_2048 • RSA_3072 • EC_P256 • EC_P256K • EC_SM2 说明 • 在中国内地使用托管密码机创建的密钥，默认为Aliyun_SM4类型，其余情况下默认为Aliyun_AES_256。 • 仅专属KMS支持Aliyun_AES_128和Aliyun_AES_192。
KeyUsage	String	否	ENCRYPT/DECRYPT	密钥的用途，取值： • ENCRYPT/DECRYPT：数据加密和解密。 • SIGN/VERIFY：产生和验证数字签名。
Origin	String	否	Aliyun_KMS	密钥材料来源，取值： • Aliyun_KMS（默认值） • EXTERNAL 说明 • 请注意区分大小写。 • 当KeySpec为非对称密钥类型时禁止选择EXTERNAL。 • 如果选择EXTERNAL，您需要导入密钥材料。
ProtectionLevel	String	否	SOFTWARE	密钥的保护级别，取值： • SOFTWARE（默认值） • HSM 说明 • 请注意区分大小写。 • 当取值为HSM时，如果Origin参数为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参数为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。
EnableAutomaticRotation	Boolean	否	false	是否开启自动密钥轮转，取值： • true • false（默认值） 说明 若Origin为EXTERNAL或KeySpec为非对称密钥类型则无法支持自动轮转。
RotationInterval	String	否	365d	自动轮转的时间周期。格式为integer[unit]，其中integer表示时间长度，unit表示时间单位。合法的unit单位为：d（天）、h（小时）、m（分钟）、s（秒）。7d或者604800s均表示7天的周期。取值范围：7~730天。 说明 当EnableAutomaticRotation参数为true时，必须设置此参数；否则，将忽略此参数。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
KeyMetadata	Struct		主密钥的Metadata。

名称	类型	示例值	描述
Arn	String	acs:kms:cn-qingdao:154035569884****:key/d6bee1cb-2e14-4277-ba6b-73786b21****	主密钥ARN。
AutomaticRotation	String	Disabled	是否开启自动密钥轮转，取值： • Enabled：自动轮转处于开启状态。 • Disabled：自动轮转处于未开启状态。 • Suspended：自动轮转被暂停执行。更多信息，请参见自动轮转密钥。 说明 仅适用于对称类型的主密钥，非对称类型的主密钥不支持自动轮转。
CreationDate	String	2016-03-25T10:42:40Z	创建主密钥的日期和时间（UTC）。
Creator	String	154035569884****	主密钥创建者。
DeleteDate	String	2020-07-06T18:22:03Z	主密钥的预计删除时间。 更多信息，请参见ScheduleKeyDeletion。 说明 只有当KeyState值为PendingDeletion时，才返回该值。
Description	String	key description example	主密钥的描述。
KeyId	String	d6bee1cb-2e14-4277-ba6b-73786b21****	主密钥的全局唯一标识符。
KeySpec	String	Aliyun_AES_256	主密钥的类型。
KeyState	String	Enabled	主密钥的状态。 更多信息，请参见用户主密钥的状态对API调用的影响。
KeyUsage	String	ENCRYPT/DECRYPT	主密钥的用途。
LastRotationDate	String	2019-06-06T18:22:03Z	最近一次轮转的时间（UTC）。 如果是新创建密钥，则为初始密钥版本生成时间。
MaterialExpireTime	String	2020-07-06T18:22:03Z	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。
NextRotationDate	String	2020-07-06T18:22:03Z	下一次轮转的时间。 说明 只有当AutomaticRotation参数值为Enabled或Suspended时，才返回该值。
Origin	String	Aliyun_KMS	主密钥的密钥材料来源。
PrimaryKeyVersion	String	7ce1d081-06cb-42e6-aab6-5c5de030****	对称类型主密钥的当前主版本标志符。 说明 • 主版本是对称类型主密钥的活跃加密密钥，密钥管理使用主版本处理加密请求。 • 不适用于非对称类型的主密钥。
ProtectionLevel	String	SOFTWARE	密钥的保护级别。
RotationInterval	String	31536000s	密钥自动轮转的周期（秒数）。格式为整数值后加上字符s。例如：7天的轮转周期为604800s。只有当AutomaticRotation参数值为Enabled或Suspended时，才返回该值。
RequestId	String	36c7e41a-3f2c-45f7-9bdd-d1dc1e7e7e06	请求ID。

示例

请求示例

```
https://[Endpoint]/?Action=CreateKey
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <KeyMetadata>
      <CreationDate>2021-04-12T06:00:54Z</CreationDate>
      <Description></Description>
      <KeyId>d6bee1cb-2e14-4277-ba6b-73786b21****</KeyId>
      <KeySpec>Aliyun_AES_256</KeySpec>
      <KeyState>Enabled</KeyState>
      <KeyUsage>ENCRYPT/DECRYPT</KeyUsage>
      <PrimaryKeyVersion>7ce1d081-06cb-42e6-aab6-5c5de030****</PrimaryKeyVersion>
      <DeleteDate></DeleteDate>
      <Creator>154035569884****</Creator>
      <Arn>acs:kms:cn-qingdao:154035569884****:key/d6bee1cb-2e14-4277-ba6b-73786b21****</Arn>
      <Origin>Aliyun_KMS</Origin>
      <MaterialExpireTime></MaterialExpireTime>
      <ProtectionLevel>SOFTWARE</ProtectionLevel>
      <LastRotationDate>2021-04-12T06:00:54Z</LastRotationDate>
      <AutomaticRotation>Disabled</AutomaticRotation>
   </KeyMetadata>
   <RequestId>36c7e41a-3f2c-45f7-9bdd-d1dc1e7e7e06</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "KeyMetadata": {
    "CreationDate": "2021-04-12T06:00:54Z",
    "Description": "",
    "KeyId": "d6bee1cb-2e14-4277-ba6b-73786b21****",
    "KeySpec": "Aliyun_AES_256",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT/DECRYPT",
    "PrimaryKeyVersion": "7ce1d081-06cb-42e6-aab6-5c5de030****",
    "DeleteDate": "",
    "Creator": "154035569884****",
    "Arn": "acs:kms:cn-qingdao:154035569884****:key/d6bee1cb-2e14-4277-ba6b-73786b21****",
    "Origin": "Aliyun_KMS",
    "MaterialExpireTime": "",
    "ProtectionLevel": "SOFTWARE",
    "LastRotationDate": "2021-04-12T06:00:54Z",
    "AutomaticRotation": "Disabled"
  },
  "RequestId": "36c7e41a-3f2c-45f7-9bdd-d1dc1e7e7e06"
}
```

错误码

访问错误中心查看更多错误码。

8.2. GetParametersForImport

调用GetParametersForImport接口获取导入主密钥材料的参数。

使用说明

- 返回的参数可用于执行ImportKeyMaterial。
- 主密钥材料来源必须是外部，即Origin为EXTERNAL。
- 本次调用返回的公钥和令牌必须搭配使用，且只能用于本次调用中指定的主密钥。
- 每次调用返回的公钥与令牌都不相同。
- 您需要指定用于加密密钥材料的公钥类型和加密算法，对应关系如下表所示。

公钥类型	加密算法	说明
RSA_2048	RSAES_PKCS1_V1_5 RSAES_OAEP_SHA_1 RSAES_OAEP_SHA_256	支持所有地域、任意保护级别的密钥。 专属KMS不支持RSAES_OAEP_SHA_1。

公钥类型	加密算法	说明
EC_SM2	SM2PKE	SM2为中国国家密码管理局批准的密码算法，仅支持导入保护级别为HSM的密钥，KMS通过部署在中国内地的托管密码机提供支持。更多信息，请参见托管密码机简介。

本文将提供一个示例，获取密钥ID为 `1234abcd-12ab-34cd-56ef-12345678****` 、加密算法为 `RSAES_PKCS1_V1_5` 、公钥类型为 `RSA_2048` 的主密钥材料参数，返回的主密钥材料参数包含密钥ID、用于加密密钥材料的公钥（PublicKey）、导入密钥材料的令牌（ImportToken）以及令牌的过期时间。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetParametersForImport	要执行的操作，取值：GetParametersForImport。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	主密钥的全局唯一标识符。 说明 密钥材料来源必须是外部，即Origin为EXTERNAL。
WrappingAlgorithm	String	是	RSAES_PKCS1_V1_5	用于加密密钥材料的算法。
WrappingKeySpec	String	是	RSA_2048	用于加密密钥材料的公钥类型。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
ImportToken	String	Base64String	导入令牌。 令牌的有效期为24小时。后续调用ImportKeyMaterial时需要指定该参数。
KeyId	String	1234abcd-12ab-34cd-56ef-12345678****	主密钥全局唯一标识符。 后续调用ImportKeyMaterial时需要指定该参数。
PublicKey	String	MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAlls4ulBxD0GG84C+lGBO6Dhpf1J3XimC6cPmPNaKKJMOzoX4tD+C+r7aZv8lZ3vnPfxuxvy/YwG+whUxTEEFUdqJTOlzhPfYucupqKM92crVHluG+xtMVeHKjyTr+UrtKCsQikqHT+19yDRN/RMoo2HUx0gmEnRyXd8t3JyUXun9FdoxKA08GrsV7nodb9ZsoBLhnev7tTLcXvLyKW6XG1ZQCQm6dPnbnwLeDXR7uK0Lqn9PM28mBldaiQUQxj2XbM1CoJA+JiyVX3Ptdb+4rqukb4Rb05B80Bs9xV/cf7Flku08l7xGhrGiQFq+DFXwQWtwihXHZxz3LhldU+4ZPwID****	用于加密密钥材料的公钥。 采用Base64编码。
RequestId	String	8cdf51fd-bcd6-d79a-0ef4-e52c9b5466dc	请求ID。
TokenExpireTime	String	2018-01-25T00:01:02Z	导入令牌的过期时间。

示例

请求示例

```
https://[Endpoint]/?Action=GetParametersForImport
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&WrappingAlgorithm=RSAES_PKCS1_V1_5
&WrappingKeySpec=RSA_2048
&<公共请求参数>
```

正常返回示例

XML 格式

```
<KMS>
    <ImportToken>Base64String</ImportToken>
    <PublicKey>MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAlls4uIBxD0GG84C+lGBO6Dhpf1J3XimC6cPmPNaKKJMOzoX4tD+C+r7aZv8lZ3vnPfxuxvy/YwG+whUxTEEFUdqJTO
IzhPfYucupqKM92crVHIuG+xtMVeHKjyTr+UrtKCsQikqHT+19yDRN/RMoo2HUx0gmEnRyXd8t3JyUXun9FdoxKA08GrsV7nodb9ZsoBLhnev7tTLcXvLyKW6XG1ZQCQm6dPnbnwLeDXR7uK0Lqn9
PM28mBIdaiQUQxj2XbM1CoJA+JiyVX3Ptdb+4rqukb4Rb05B80Bs9xV/cf7FIku08l7xGhrGiQFq+DFXwQWtwihXHZxz3LhldU+4ZPwID****</PublicKey>
    <KeyId>1234abcd-12ab-34cd-56ef-12345678****</KeyId>
    <TokenExpireTime>2018-01-25T00:01:02Z</TokenExpireTime>
    <RequestId>8cdf51fd-bcd6-d79a-0ef4-e52c9b5466dc</RequestId>
</KMS>
```

JSON 格式

```
{
        "ImportToken":"Base64String",
        "PublicKey":"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAlls4uIBxD0GG84C+lGBO6Dhpf1J3XimC6cPmPNaKKJMOzoX4tD+C+r7aZv8lZ3vnPfxuxvy/YwG+whUxTEEF
UdqJTOIzhPfYucupqKM92crVHIuG+xtMVeHKjyTr+UrtKCsQikqHT+19yDRN/RMoo2HUx0gmEnRyXd8t3JyUXun9FdoxKA08GrsV7nodb9ZsoBLhnev7tTLcXvLyKW6XG1ZQCQm6dPnbnwLeDXR7u
K0Lqn9PM28mBIdaiQUQxj2XbM1CoJA+JiyVX3Ptdb+4rqukb4Rb05B80Bs9xV/cf7FIku08l7xGhrGiQFq+DFXwQWtwihXHZxz3LhldU+4ZPwID****",
        "KeyId":"1234abcd-12ab-34cd-56ef-12345678****",
        "TokenExpireTime":"2018-01-25T00:01:02Z",
        "RequestId":"8cdf51fd-bcd6-d79a-0ef4-e52c9b5466dc"
}
```

错误码

访问错误中心查看更多错误码。

8.3. ImportKeyMaterial

调用ImportKeyMaterial接口导入密钥材料。

调用CreateKey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**。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DescribeKey。
- 在导入密钥材料之前， 需要调用GetParametersForImport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（PublicKey）和导入令牌（ImportToken）。

> **说明**
> - 对密钥类型为**Aliyun_AES_256**的CMK，密钥材料必须为256位；对密钥类型为**Aliyun_SM4**的CMK，密钥材料必须为128位。
> - 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。
> - 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加密的数据或生成的数据密钥（Data Key）无法使用另一个CMK解密。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ImportKeyMaterial	要执行的操作，取值：**ImportKeyMaterial**。
EncryptedKeyMaterial	String	是	bCPZx7l6v6KXsqEpr2OXKxuj2CCRtKdwp75Bw+BGncYqBdfjFBYRtOE6HRlT0oeiRDWzwnw9OA54OL36smDJrq4Lo9x0CyYDiuKnRkcKtMtlzW0din7Pd7llZWWRdVueiw2qpzl7PkUWQGTdsdbzpfJJQ+qj/cRlrk/E83UGyeyytSpgnb+lu0xEYcPajRyWNsbi98N3pqqQzHXNNHO2NJqHlnQgglqTiBEjkGeKFhfKmTc3vjulldVa3EaVIN6lwWfgx+UUYSrvbA77WDYKlDsZ4SbK2/T7za9Tp1qU7Ynqba7OKGVVj7PMbiaO80AxWZnjUMYCgEp5w7V+seOXqw==	使用**GetParametersForImport**返回的公钥加密并用base64编码后的密钥材料。
ImportToken	String	是	Base64String	通过调用**GetParametersForImport**获得的导入令牌。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	待导入的主密钥ID。

名称	类型	是否必选	示例值	描述
KeyMaterialExpireUnix	Long	是	0	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。 说明 取值不可早于调用该API的时间（以服务器时间为准）。

返回数据

名称	类型	示例值	描述
RequestId	String	ec1017cf-ead4-f3ca-babc-c3b34f3dbecb	请求ID。

示例

请求示例

```
https://[Endpoint]/?Action=ImportKeyMaterial
&EncryptedKeyMaterial=bCPZx7I6v6KXsqEpr2OXKxuj2CCRtKdwp75Bw+BGncYqBdfjFBYRtOE6HRlT0oeiRDWzwnw9OA54OL36smDJrq4Lo9x0CyYDiuKnRkcKtMtlzW0din7Pd7IlZWWRdVueiw2qpzl7PkUWQGTdsdbzpfJJQ+qj/cRIrk/E83UGyeyytSpgnb+lu0xEYcPajRyWNsbi98N3pqqQzHXNNHO2NJqHlnQgglqTiBEjkGeKFhfKmTc3vjulIdVa3EaVIN6lwWfgx+UUYSrvbA77WDYKlDsZ4SbK2/T7za9Tp1qU7Ynqba7OKGVVj7PMbiaO80AxWZnjUMYCgEp5w7V+seOXqw==
&ImportToken=Base64String
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&KeyMaterialExpireUnix=0
&<公共请求参数>
```

正常返回示例

`XML` 格式

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

`JSON` 格式

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

错误码

访问错误中心查看更多错误码。

8.4. EnableKey

调用EnableKey接口启用指定的主密钥进行加解密。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	EnableKey	要执行的操作，取值：EnableKey。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	密钥ID。主密钥的全局唯一标识符。

返回数据

名称	类型	示例值	描述
RequestId	String	efb1cbbd-a093-4278-bc03-639dd4fcc207	请求ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=EnableKey
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

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

`JSON` 格式

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

错误码

访问错误中心查看更多错误码。

8.5. DisableKey

调用DisableKey接口禁用指定的主密钥（CMK）进行加解密。

使用说明：禁用主密钥后，原来使用该主密钥加密的密文无法解密。您可以调用EnableKey将主密钥恢复至启用状态，然后解密密文。

本文将提供一个示例，禁用ID为 `1234abcd-12ab-34cd-56ef-12345678****` 的主密钥进行加解密。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DisableKey	要执行的操作，取值：DisableKey。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	密钥ID。主密钥的全局唯一标识符。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
RequestId	String	2fe70ce2-3303-4fd6-b3ac-472fb2705c62	请求ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=DisableKey
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

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

`JSON` 格式

```
{
 "RequestId": "2fe70ce2-3303-4fd6-b3ac-472fb2705c62"
}
```

错误码

访问错误中心查看更多错误码。

8.6. SetDeletionProtection

调用SetDeletionProtection接口为用户主密钥（CMK）开启或关闭删除保护。

使用说明：

- 当您为CMK开启删除保护后，将无法删除该CMK。如果需要删除CMK，需提前关闭删除保护。
- 调用SetDeletionProtection接口前，请确保CMK不处于待删除状态。您可以调用DescribeKey接口查看CMK的状态（KeyState）。

本文将提供一个示例，为CMK ARN（ProtectedResourceArn）为 `acs:kms:cn-hangzhou:123213123****:key/0225f411-b21d-46d1-be5b-93931c82****` 的CMK开启删除保护。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	SetDeletionProtection	要执行的操作，取值：SetDeletionProtection。
EnableDeletionProtection	Boolean	是	true	是否开启删除保护，取值： • true：开启删除保护。 • false（默认值）：关闭删除保护。
ProtectedResourceArn	String	是	acs:kms:cn-hangzhou:123213123****:key/0225f411-b21d-46d1-be5b-93931c82****	待设置删除保护的CMK ARN。 您可以调用DescribeKey接口查看CMK ARN（Arn）。
DeletionProtectionDescription	String	否	该密钥正在被XXX服务使用。已为您设置删除保护。	删除保护描述。 说明 当EnableDeletionProtection取值为true时该参数有效。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
RequestId	String	3455b9b4-95c1-419d-b310-db6a53b09a39	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=SetDeletionProtection
&EnableDeletionProtection=true
&ProtectedResourceArn=acs:kms:cn-hangzhou:123213123****:key/0225f411-b21d-46d1-be5b-93931c82****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
      <RequestId>3455b9b4-95c1-419d-b310-db6a53b09a39</RequestId>
</KMS>
```

`JSON` 格式

```
{
    "RequestId":"3455b9b4-95c1-419d-b310-db6a53b09a39"
}
```

错误码

访问错误中心查看更多错误码。

8.7. ScheduleKeyDeletion

调用ScheduleKeyDeletion接口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状态，无法用于加密、解密、产生数据密钥操作。

主密钥一旦删除，将无法恢复，使用该主密钥加密的内容及产生的数据密钥也将无法解密。因此，对于主密钥的删除，KMS只提供申请删除的方式，而不提供直接删除的方式。如果您有删除密钥方面的需求，可以通过DisableKey禁用密钥。

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。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ScheduleKeyDeletion	要执行的操作，取值：ScheduleKeyDeletion。

名称	类型	是否必选	示例值	描述
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	密钥ID。CMK全局唯一标识符。
PendingWindowInDays	Integer	否	7	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。 取值范围：7~30。 单位：天。

返回数据

名称	类型	示例值	描述
RequestId	String	3da5b8cc-8107-40ac-a170-793cd181d7b7	请求ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=ScheduleKeyDeletion
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

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

`JSON` 格式

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

错误码

HttpCode	错误码	错误信息	描述
400	Throttling	Request was denied due to request throttling.	您这个时段的流量已经超限。如果不能满足现有业务要求可以提工单进行申请。
404	Forbidden.KeyNotFound	The specified Key is not found.	指定的密钥不存在。

访问错误中心查看更多错误码。

8.8. CancelKeyDeletion

调用CancelKeyDeletion接口撤销密钥删除。

当密钥删除的申请撤销成功以后，密钥会处于启用状态。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CancelKeyDeletion	要执行的操作，取值：CancelKeyDeletion。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	主密钥的全局唯一标识符。

返回数据

名称	类型	示例值	描述
RequestId	String	3da5b8cc-8107-40ac-a170-793cd181d7b7	本次请求的ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CancelKeyDeletion
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

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

`JSON` 格式

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

错误码

访问错误中心查看更多错误码。

8.9. DeleteKeyMaterial

调用DeleteKeyMaterial接口删除已导入的密钥材料。

此操作不会删除密钥材料对应的主密钥（CMK）。

如果主密钥处于待删除状态，删除密钥材料不会改变密钥状态和预计删除时间；如果主密钥不处于待删除状态，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，您可以重新导入密钥材料，但必须与之前的密钥材料相同。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteKeyMaterial	要执行的操作，取值：DeleteKeyMaterial。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	密钥ID。主密钥（CMK）的全局唯一标识符。

返回数据

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

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=DeleteKeyMaterial
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

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

`JSON` 格式

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

错误码

访问错误中心查看更多错误码。

8.10. DescribeKey

调用DescribeKey接口查询用户主密钥（CMK）详情。

本文将提供一个示例，为您查询ID为 `05754286-3ba2-4fa6-8d41-4323aca6****` 的用户主密钥的详情，包括创建者、创建时间、密钥状态、删除保护状态等信息。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeKey	要执行的操作，取值：DescribeKey。
KeyId	String	是	05754286-3ba2-4fa6-8d41-4323aca6****	CMK的全局唯一标识符。 该参数也可以被指定为CMK绑定的别名。更多信息，请参见别名概述。

返回数据

名称	类型	示例值	描述
KeyMetadata	Struct		CMK的元数据。
Arn	String	acs:kms:cn-hangzhou:154035569884****:key/05754286-3ba2-4fa6-8d41-4323aca6****	CMK ARN。
AutomaticRotation	String	Disabled	是否开启自动轮转，取值： • Enabled：开启自动轮转。 • Disabled：关闭自动轮转。 • Suspended：暂停执行自动轮转。 更多信息，请参见自动轮转密钥。 说明 仅对称密钥支持自动轮转。
CreationDate	String	2021-05-20T06:34:21Z	CMK的创建时间（UTC）。
Creator	String	154035569884****	CMK的创建者（阿里云账号）。
DeleteDate	String	2021-05-26T18:22:03Z	CMK的预计删除时间（UTC）。 更多信息，请参见ScheduleKeyDeletion。 说明 当KeyState取值为PendingDeletion时，返回该参数。
DeletionProtection	String	Enabled	是否开启删除保护，取值： • Enabled：开启删除保护。 • Disabled：关闭删除保护。
DeletionProtectionDescription	String	该密钥正在被XXX服务使用。已为您设置删除保护。	删除保护描述。
Description	String	key description example	CMK的描述。
KeyId	String	05754286-3ba2-4fa6-8d41-4323aca6****	CMK的全局唯一标识符。
KeySpec	String	Aliyun_AES_256	CMK的类型。
KeyState	String	Enabled	CMK的状态。 更多信息，请参见用户主密钥的状态对API调用的影响。
KeyUsage	String	ENCRYPT/DECRYPT	CMK的用途。
LastRotationDate	String	2021-05-20T06:34:21Z	最近一次轮转的时间（UTC）。如果是新创建的密钥，则为初始密钥版本生成时间。
MaterialExpireTime	String	2021-07-06T18:22:03Z	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。

名称	类型	示例值	描述
NextRotationDate	String	2021-07-06T18:22:03Z	下一次轮转的时间。 说明 当AutomaticRotation取值为Enabled或Suspended时，返回该参数。
Origin	String	Aliyun_KMS	CMK的密钥材料来源。
PrimaryKeyVersion	String	515e0b0a-624f-45ab-92b5-54f9b551****	对称类型CMK当前的主版本标识符。
ProtectionLevel	String	HSM	密钥的保护级别。
RotationInterval	String	31536000s	密钥自动轮转的周期。 单位：s。 例如：7天的轮转周期为604800s。 说明 当AutomaticRotation取值为Enabled或Suspended时，返回该参数。
RequestId	String	f1fdfa9d-bd49-418b-942f-8f3e3ec00a4f	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DescribeKey
&KeyId=05754286-3ba2-4fa6-8d41-4323aca6****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <KeyMetadata>
      <CreationDate>2021-05-20T06:34:21Z</CreationDate>
      <Description></Description>
      <KeyId>05754286-3ba2-4fa6-8d41-4323aca6****</KeyId>
      <KeySpec>Aliyun_AES_256</KeySpec>
      <KeyState>Enabled</KeyState>
      <KeyUsage>ENCRYPT/DECRYPT</KeyUsage>
      <PrimaryKeyVersion>515e0b0a-624f-45ab-92b5-54f9b551****</PrimaryKeyVersion>
      <DeleteDate></DeleteDate>
      <Creator>154035569884****</Creator>
      <Arn>acs:kms:cn-hangzhou:154035569884****:key/05754286-3ba2-4fa6-8d41-4323aca6****</Arn>
      <Origin>Aliyun_KMS</Origin>
      <MaterialExpireTime></MaterialExpireTime>
      <ProtectionLevel>HSM</ProtectionLevel>
      <LastRotationDate>2021-05-20T06:34:21Z</LastRotationDate>
      <AutomaticRotation>Disabled</AutomaticRotation>
      <DeletionProtection>Enabled</DeletionProtection>
   </KeyMetadata>
   <RequestId>f1fdfa9d-bd49-418b-942f-8f3e3ec00a4f</RequestId>
</KMS>
```

`JSON` 格式

```
{
 "KeyMetadata": {
  "CreationDate": "2021-05-20T06:34:21Z",
  "Description": "",
  "KeyId": "05754286-3ba2-4fa6-8d41-4323aca6****",
  "KeySpec": "Aliyun_AES_256",
  "KeyState": "Enabled",
  "KeyUsage": "ENCRYPT/DECRYPT",
  "PrimaryKeyVersion": "515e0b0a-624f-45ab-92b5-54f9b551****",
  "DeleteDate": "",
  "Creator": "1540355698884****",
  "Arn": "acs:kms:cn-hangzhou:1540355698884****:key/05754286-3ba2-4fa6-8d41-4323aca6****",
  "Origin": "Aliyun_KMS",
  "MaterialExpireTime": "",
  "ProtectionLevel": "HSM",
  "LastRotationDate": "2021-05-20T06:34:21Z",
  "AutomaticRotation": "Disabled",
  "DeletionProtection": "Enabled"
 },
 "RequestId": "f1fdfa9d-bd49-418b-942f-8f3e3ec00a4f"
}
```


错误码

HttpCode	错误码	错误信息	描述
404	Forbidden.KeyNotFound	The specified Key is not found.	指定的密钥不存在。

访问错误中心查看更多错误码。

8.11. ListKeys

调用ListKeys查询调用者在调用地域的所有主密钥ID。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListKeys	要执行的操作，取值：ListKeys。
PageNumber	Integer	否	1	当前页数。 取值范围：大于0。 默认值：1。
PageSize	Integer	否	10	每页返回值的个数。 取值范围：1~100。 默认值：10。

名称	类型	是否必选	示例值	描述

Filters	String	否	[{"Key":"KeyState", "Values":["Enabled","Disabled"]}]	主密钥过滤器。由Key-Values键值对组成，长度为0~10。
				• Key
				  ○ 描述：需要过滤的属性。
				  ○ 类型：String。
				  ○ 取值：
				  ■ KeyState：密钥状态。
				  ■ KeySpec：密钥类型。
				  ■ KeyUsage：密钥用途。
				  ■ ProtectionLevel：保护等级。
				  ■ CreatorType：创建者类型。
				• Values
				  ○ 描述：期望过滤后包含的值。
				  ○ 类型：String数组。
				  ○ 长度：0~10。
				  ○ 取值：
				  ■ Key取值为KeyState时：Enabled（启用）、Disabled（禁用）、PendingDeletion（待删除）、PendingImport（待导入）。
				  ■ Key取值为KeySpec时：Aliyun_AES_256、Aliyun_SM4、RSA_2048、EC_P256、EC_P256K、EC_SM2。
				  说明：仅在支持托管密码机且已通过国密局商用密码检测认证的地域可以创建EC_SM2和Aliyun_SM4类型的密钥，地域详情请参见支持的地域。如果您所选择地域不支持EC_SM2和Aliyun_SM4，指定这两个参数将被忽略。
				  ■ Key取值为KeyUsage时：ENCRYPT/DECRYPT（数据加密和解密）、SIGN/VERIFY（产生和验证数字签名）。
				  ■ Key取值为ProtectionLevel时：SOFTWARE（软件）、HSM（硬件）。
				  说明：HSM保护等级仅在特定地域支持，地域详情请参见支持的地域。如您所选择地域不支持HSM，指定该参数将被忽略。
				  ■ Key取值为CreatorType时：User（获取由用户创建的主密钥）、Service（获取由用户授权其他云产品自动创建的主密钥）。
				Filters不同Key之间的逻辑关系为AND，同一个Key中的多个Value之间的逻辑关系为OR。例如：输入
				`[{"Key":"KeyState", "Values":["Enabled","Disabled"]}, {"Key":"KeyState", "Values":["PendingDeletion"]}, {"Key":"KeySpec", "Values":["Aliyun_AES_256"]}]` 时，语义为：`(KeyState=Enabled OR KeyState=Disabled OR KeyState=PendingDeletion) AND (KeySpec=Aliyun_AES_256)`。

返回数据

名称	类型	示例值	描述
Keys	Array of Key		主密钥。
Key			
KeyArn	String	acs:kms:cn-hangzhou:123456:key/80e9409f-78fa-42ab-84bd-83f40c81****	主密钥的ARN。
KeyId	String	08c33a6f-4e0a-4a1b-a3fa-7ddfa1d4****	主密钥的全局唯一标识符。
PageNumber	Integer	1	当前页数。
PageSize	Integer	10	每页返回值的个数。

名称	类型	示例值	描述
RequestId	String	1050b8f1-b264-496d-a782-6299cbaf15f8	请求ID。
TotalCount	Integer	3	主密钥的总数。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=ListKeys
&PageNumber=1
&PageSize=10
&Filters=[{"Key":"KeyState", "Values":["Enabled","Disabled"]}]
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
          <Keys>
                    <Key>
                              <KeyId>80e9409f-78fa-42ab-84bd-83f40c81****</KeyId>
                              <KeyArn>acs:kms:cn-hangzhou:123456:key/80e9409f-78fa-42ab-84bd-83f40c81****</KeyArn>
                    </Key>
                    <Key>
                              <KeyId>8fbb1226-a06f-4c57-8887-daa5b627****</KeyId>
                              <KeyArn>acs:kms:cn-hangzhou:123456:key/8fbb1226-a06f-4c57-8887-daa5b627****</KeyArn>
                    </Key>
                      </Keys>
          <TotalCount>3</TotalCount>
          <PageNumber>1</PageNumber>
          <PageSize>10</PageSize>
          <RequestId>bc29ec35-3373-48d1-8202-520fd78d****</RequestId>
</KMS>
```

`JSON` 格式

```
{
        "Keys": {
                "Key": [
                        {
                                "KeyId": "80e9409f-78fa-42ab-84bd-83f40c81****",
                    "KeyArn": "acs:kms:cn-hangzhou:123456:key/80e9409f-78fa-42ab-84bd-83f40c81****"
                        },
                        {
                                "KeyId": "8fbb1226-a06f-4c57-8887-daa5b627****",
                    "KeyArn": "acs:kms:cn-hangzhou:123456:key/8fbb1226-a06f-4c57-8887-daa5b627****"
                        }
                    ],
                "TotalCount": 3,
        "PageNumber": 1,
        "PageSize": 10,
        "RequestId": "bc29ec35-3373-48d1-8202-520fd78d****"
}
```

错误码

访问错误中心查看更多错误码。

8.12. UpdateKeyDescription

调用UpdateKeyDescription接口更新主密钥的描述信息。

将主密钥（CMK）的描述信息（DescribeKey接口中的Description属性）替换为用户传入的值。使用此API可以对密钥的描述信息进行添加、变更、删除操作。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateKeyDescription	要执行的操作，取值：UpdateKeyDescription。
Description	String	是	key description example	主密钥的描述性信息。通常用于描述主密钥的用途，例如主密钥保护的数据类型、可使用主密钥的应用等。

名称	类型	是否必选	示例值	描述
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	密钥ID。主密钥的全局唯一标识符。

返回数据

名称	类型	示例值	描述
RequestId	String	3455b9b4-95c1-419d-b310-db6a53b09a39	请求ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=UpdateKeyDescription
&Description=key description example
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <RequestId>3455b9b4-95c1-419d-b310-db6a53b09a39</RequestId>
</KMS>
```

`JSON` 格式

```
{
 "RequestId": "3455b9b4-95c1-419d-b310-db6a53b09a39"
}
```

错误码

访问错误中心查看更多错误码。

8.13. DescribeKeyVersion

调用DescribeKeyVersion接口查询指定密钥版本信息。

本文将提供一个示例，为您查询密钥 `1234abcd-12ab-34cd-56ef-12345678****` 的密钥版本信息，密钥版本ID为 `2ab1a983-7072-4bbc-a582-584b5bd8****` 。返回结果显示，密钥版本的创建时间为 `2016-03-25T10:42:40Z` 。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeKeyVersion	要执行的操作，取值：DescribeKeyVersion。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	主密钥（CMK）的全局唯一标识符。 该参数也可以被指定为CMK绑定的别名。更多信息，请参见别名使用说明。
KeyVersionId	String	是	2ab1a983-7072-4bbc-a582-584b5bd8****	密钥版本的全局唯一标识符。 您可以调用ListKeyVersions接口，获取KeyVersionId。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
KeyVersion	Struct		密钥版本的元数据。
CreationDate	String	2016-03-25T10:42:40Z	创建密钥版本时的日期和时间（UTC时间）。

名称	类型	示例值	描述
KeyId	String	1234abcd-12ab-34cd-56ef-12345678****	CMK的全局唯一标识符。 说明 如果请求中的KeyId参数使用的是CMK的别名，在响应中会返回别名对应的CMK标识符。
KeyVersionId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	密钥版本的全局唯一标识符。
RequestId	String	7021b6ec-4be7-4d3c-8a68-1e85d4d515a0	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DescribeKeyVersion
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&KeyVersionId=2ab1a983-7072-4bbc-a582-584b5bd8****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <RequestId>7021b6ec-4be7-4d3c-8a68-1e85d4d515a0</RequestId>
        <KeyVersion>
                <CreationDate>2016-03-25T10:42:40Z</CreationDate>
                <KeyId>1234abcd-12ab-34cd-56ef-12345678****</KeyId>
                <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
        </KeyVersion>
</KMS>
```

`JSON` 格式

```
{
    "RequestId":"7021b6ec-4be7-4d3c-8a68-1e85d4d515a0",
    "KeyVersion":{
        "CreationDate":"2016-03-25T10:42:40Z",
        "KeyId":"1234abcd-12ab-34cd-56ef-12345678****",
        "KeyVersionId":"2ab1a983-7072-4bbc-a582-584b5bd8****"
        }
}
```

错误码

访问错误中心查看更多错误码。

8.14. ListKeyVersions

调用ListKeyVersions接口列出主密钥的所有密钥版本。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListKeyVersions	要执行的操作，取值：ListKeyVersions。
KeyId	String	是	0b30658a-ed1a-4922-b8f7-a673ca9c****	主密钥（CMK）的全局唯一标识符。该参数也可以被指定为CMK绑定的别名，详情请参见别名使用说明。
PageNumber	Integer	是	1	当前页数。 取值为大于0的整数。 默认值：1。
PageSize	Integer	否	10	每页返回的结果个数。 取值范围：0~101。 默认值：10。

返回数据

名称	类型	示例值	描述
KeyVersions	Array		返回的密钥版本数组。
KeyVersion			
CreationDate	String	2016-03-25T10:42:40Z	创建密钥版本时的日期和时间（UTC时间）。
KeyId	String	0b30658a-ed1a-4922-b8f7-a673ca9c****	CMK的全局唯一标识符。 说明 如果请求中的KeyId参数使用的是CMK的别名，在响应中会返回别名对应的CMK标识符。
KeyVersionId	String	1e3304fd-68ac-4d5b-8886-ae5f01a1****	密钥版本的全局唯一标识符。
PageNumber	Integer	1	当前页数。
PageSize	Integer	10	每页的返回结果个数。
RequestId	String	f71204c4-53cd-4eea-b405-653ba2db7e86	本次请求的ID。
TotalCount	Integer	3	返回的密钥版本总数。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ListKeyVersions
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&PageNumber=1
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <RequestId>f71204c4-53cd-4eea-b405-653ba2db7e86</RequestId>
        <KeyVersions>
                <KeyVersion>
                        <KeyId>0b30658a-ed1a-4922-b8f7-a673ca9c****</KeyId>
                        <KeyVersionId>1e3304fd-68ac-4d5b-8886-ae5f01a1****</KeyVersionId>
                        <CreationDate>2019-08-06T10:22:03Z</CreationDate>
                </KeyVersion>
                <KeyVersion>
                        <KeyId>0b30658a-ed1a-4922-b8f7-a673ca9c****</KeyId>
                        <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
                        <CreationDate>2019-08-06T10:19:18Z</CreationDate>
                </KeyVersion>
                <KeyVersion>
                        <KeyId>0b30658a-ed1a-4922-b8f7-a673ca9c****</KeyId>
                        <KeyVersionId>6a69c763-388a-4708-9fc0-4322266b****</KeyVersionId>
                        <CreationDate>2019-08-06T10:17:04Z</CreationDate>
                </KeyVersion>
        </KeyVersions>
        <TotalCount>3</TotalCount>
        <PageNumber>1</PageNumber>
        <PageSize>10</PageSize>
</KMS>
```

`JSON` 格式

```
{
    "KMS": {
        "RequestId": "f71204c4-53cd-4eea-b405-653ba2db7e86",
        "KeyVersions": {
            "KeyVersion": [
                {
                    "KeyId": "0b30658a-ed1a-4922-b8f7-a673ca9c****",
                    "KeyVersionId": "1e3304fd-68ac-4d5b-8886-ae5f01a1****",
                    "CreationDate": "2019-08-06T10:22:03Z"
                },
                {
                    "KeyId": "0b30658a-ed1a-4922-b8f7-a673ca9c****",
                    "KeyVersionId": "2ab1a983-7072-4bbc-a582-584b5bd8****",
                    "CreationDate": "2019-08-06T10:19:18Z"
                },
                {
                    "KeyId": "0b30658a-ed1a-4922-b8f7-a673ca9c****",
                    "KeyVersionId": "6a69c763-388a-4708-9fc0-4322266b****",
                    "CreationDate": "2019-08-06T10:17:04Z"
                }
            ]
        },
        "TotalCount": 3,
        "PageNumber": 1,
        "PageSize": 10
    }
}
```


错误码

访问错误中心查看更多错误码。

8.15. UpdateRotationPolicy

更新密钥轮转策略。

如果开启自动轮转，将在上一次轮转时间加上轮转周期天数后，自动创建一个新的密钥版本，并将它设置为主版本。下列情况不允许配置自动轮转策略：

- 指定的主密钥为非对称密钥。
- 指定的主密钥为云产品托管的默认密钥。
- 指定的主密钥为用户自带密钥（外部导入到KMS的密钥）。
- 指定的主密钥处于**Enabled**之外的状态。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateRotationPolicy	系统规定参数。取值：UpdateRotationPolicy。
EnableAutomaticRotation	Boolean	是	true	是否开启自动密钥轮转。取值：true | false。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	主密钥的全局唯一标识符。
RotationInterval	String	否	30d	自动轮转的时间周期。格式为integer[unit]，其中integer表示时间长度，unit表示时间单位。合法的unit单位为：d（天）、h（小时）、m（分钟）、s（秒）。**7d**或者**604800s**均表示7天的周期。取值：7~730天。 说明 当**EnableAutomaticRotation**参数为true时，必须设置此参数；反之，将忽略此参数。

返回数据

名称	类型	示例值	描述
RequestId	String	efb1cbbd-a093-4278-bc03-639dd4fcc207	本次请求的ID。

示例

请求示例

```
https://[Endpoint]/?Action=UpdateRotationPolicy
&EnableAutomaticRotation=true
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&RotationInterval=30d
&<公共请求参数>
```

正常返回示例

`XML` 格式

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

`JSON` 格式

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

错误码

访问错误中心查看更多错误码。

8.16. CreateKeyVersion

调用CreateKeyVersion接口为用户主密钥（CMK）创建密钥版本。

使用限制：

- 您只能为非对称类型的CMK创建密钥版本，且CMK必须处于开启（Enabled）状态。您可以调用CreateKey 接口创建非对称密钥，调用DescribeKey 接口查询密钥状态（KeyState）。
- 创建密钥版本的最小间隔为7天。您可以调用DescribeKey 接口查询上次创建密钥版本的时间（LastRotationDate）。
- PrivateKeyStore中的CMK不支持创建密钥版本。
- 每个地域最多支持创建50个非对称密钥版本。

本文将提供一个示例，为ID为 `0b30658a-ed1a-4922-b8f7-a673ca9c****` 的CMK创建密钥版本。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateKeyVersion	要执行的操作，取值：CreateKeyVersion。
KeyId	String	是	0b30658a-ed1a-4922-b8f7-a673ca9c****	用户主密钥（CMK）的全局唯一标识符。 说明 该参数也可以被指定为CMK绑定的别名。更多信息，请参见别名概述。

返回数据

名称	类型	示例值	描述
KeyVersion	Struct		密钥版本的元数据。
CreationDate	String	2019-08-02T10:38:27Z	创建密钥版本的时间（UTC时间）。
KeyId	String	0b30658a-ed1a-4922-b8f7-a673ca9c****	用户主密钥（CMK）的全局唯一标识符。
KeyVersionId	String	c0a3d5dc-0b47-4199-a050-b289349a****	密钥版本的标识符。
RequestId	String	b96f250a-4b75-498c-91be-22c6928f85be	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CreateKeyVersion
&KeyId=0b30658a-ed1a-4922-b8f7-a673ca9c****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
    <KeyVersion>
        <KeyId>0b30658a-ed1a-4922-b8f7-a673ca9c****</KeyId>
        <KeyVersionId>c0a3d5dc-0b47-4199-a050-b289349a****</KeyVersionId>
        <CreationDate>2019-08-02T10:38:27Z</CreationDate>
    </KeyVersion>
    <RequestId>b96f250a-4b75-498c-91be-22c6928f85be</RequestId>
</KMS>
```

`JSON` 格式

```
{
        "KeyVersion": {
                "KeyId": "0b30658a-ed1a-4922-b8f7-a673ca9c****",
                "KeyVersionId": "c0a3d5dc-0b47-4199-a050-b289349a****",
                "CreationDate": "2019-08-02T10:38:27Z"
        },
        "RequestId": "b96f250a-4b75-498c-91be-22c6928f85be"
}
```

错误码

访问错误中心查看更多错误码。

8.17. Encrypt

调用Encrypt接口使用对称主密钥（Symmetric CMK）将明文加密为密文。

- KMS使用指定CMK的主版本对传入数据进行加密。
- 最多可加密6KB的数据，例如RSA密钥、数据库密码或其它敏感信息。
- 如果将加密数据从一个地域迁移到另一个地域，可以调用Encrypt接口在新地域中加密从另一个地域中转移过来的明文DataKey。新地域中会生成一个加密后的DataKey。您可以在新地域调用Decrypt将其解密。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	Encrypt	要执行的操作，取值：Encrypt。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	主密钥（CMK）的全局唯一标识符。该参数也可以被指定为CMK绑定的别名，详情请参见别名使用说明。
Plaintext	String	是	SGVsbG8gd29y****	待加密明文（必须经过Base64编码）。
EncryptionContext	Json	否	{"Example":"Example"}	key/value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，详情请参见EncryptionContext说明。

返回数据

名称	类型	示例值	描述
CiphertextBlob	String	DZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmaaSl+TztSlMe43nbTH/Z1Wr4XfLftKhAciUmDQXuMRl4WTvKhxjMThjK****	数据被指定CMK的主版本加密后的密文。
KeyId	String	1234abcd-12ab-34cd-56ef-12345678****	CMK的全局唯一标识符。如果请求中的KeyId参数使用的是CMK的别名，在响应中会返回别名对应的CMK标志符。
KeyVersionId	String	86a9efd9-3d16-4894-bd4f-1fc43f3f****	用于加密明文的密钥版本标志符。是指定CMK的主版本。
RequestId	String	475f1620-b9d3-4d35-b5c6-3fbdd941423d	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=Encrypt
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&Plaintext=SGVsbG8gd29y****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <RequestId>475f1620-b9d3-4d35-b5c6-3fbdd941423d</RequestId>
        <CiphertextBlob>DZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmaaSl+TztSIMe43nbTH/Z1Wr4XfLftKhAciUmDQXuMRl4WTvKhxjMThjK****</CiphertextBlob>
        <KeyId>1234abcd-12ab-34cd-56ef-12345678****</KeyId>
        <KeyVersionId>86a9efd9-3d16-4894-bd4f-1fc43f3f****</KeyVersionId>
</KMS>
```

`JSON` 格式

```
{
    "RequestId":"475f1620-b9d3-4d35-b5c6-3fbdd941423d",
    "CiphertextBlob":"DZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmaaSl+TztSIMe43nbTH/Z1Wr4XfLftKhAciUmDQXuMRl4WTvKhxjMThjK****",
    "KeyId":"1234abcd-12ab-34cd-56ef-12345678****",
    "KeyVersionId":"86a9efd9-3d16-4894-bd4f-1fc43f3f****"
}
```

错误码

访问错误中心查看更多错误码。

8.18. GenerateDataKey

调用GenerateDataKey接口生成一个随机的数据密钥，用于本地数据加密。

API随机生成的数据密钥通过您指定的主密钥（CMK）加密后，返回数据密钥的密文和明文。您可以使用返回的数据密钥明文，在KMS之外对数据进行本地离线加密。在存储加密后的数据时，也需要存储数据密钥的密文。您可以通过响应中的Plaintext字段获取到数据密钥的明文，通过CiphertextBlob字段获取到数据密钥的密文。

在请求中指定的CMK，仅会被用作数据密钥的加密，和数据密钥的生成无关。KMS不会记录或存储随机生成的数据密钥，您需要负责对数据密钥（密文）进行持久化。

建议您使用以下方式在本地进行数据加密：

1. 调用GenerateDataKey接口，获得数据加密密钥。
2. 使用数据密钥的明文（通过响应中的Plaintext字段返回），在本地完成离线数据加密，随后清除内存中的数据密钥明文。
3. 将数据密钥的密文（通过响应中的CiphertextBlob字段返回），和本地离线加密后的数据一并进行存储。

在本地解密数据：

- 调用Decrypt接口解密本地存储的数据密钥的密文。这一操作将返回数据密钥的明文。
- 使用数据密钥的明文，在本地完成离线数据解密，随后清除内存中的数据密钥明文。

本文将提供一个示例，为ID为 `1234abcd-12ab-34cd-56ef-12345678****` 的密钥生成随机的数据密钥。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GenerateDataKey	要执行的操作，取值：GenerateDataKey。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	主密钥（CMK）的全局唯一标识符。 该参数也可以被指定为主密钥绑定的别名。更多信息，请参见别名使用说明。
KeySpec	String	否	AES_256	指定生成的数据密钥的长度，取值： • AES_256：256比特的对称密钥。 • AES_128：128比特的对称密钥。 **说明** 建议使用KeySpec或者NumberOfBytes来指定数据密钥长度。如果两者都不指定，KMS生成256比特的数据密钥；如果两者都被指定，KMS会忽略KeySpec参数。
NumberOfBytes	Integer	否	256	指定生成的数据密钥的长度。 取值：1~1024。 单位：字节。
EncryptionContext	Json	否	{"Example":"Example"}	key/value对的JSON字符串。 如果指定了该参数，则在调用Decrypt接口时需要提供同样的参数。更多信息，请参见EncryptionContext。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
CiphertextBlob	String	ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901qOjop4bTS****	数据密钥被指定CMK的主版本加密后的密文。
KeyId	String	599fa825-17de-417e-9554-bb032cc6****	主密钥的全局唯一标识符。 说明 如果请求中的KeyId参数使用的是CMK的别名，在响应中会返回别名对应的CMK标识符。
KeyVersionId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	密钥版本ID。主密钥版本的全局唯一标识符。
Plaintext	String	QmFzZTY0IGVuY29kZWQgcGxhaW50ZXh0	数据密钥的明文经过Base64编码的后的值。
RequestId	String	7021b6ec-4be7-4d3c-8a68-1e85d4d515a0	请求ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=GenerateDataKey
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
         <CiphertextBlob>ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901qOjop4bTS****</CiphertextBlob>
         <KeyId>599fa825-17de-417e-9554-bb032cc6****</KeyId>
         <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
         <Plaintext>QmFzZTY0IGVuY29kZWQgcGxhaW50ZXh0</Plaintext>
         <RequestId>7021b6ec-4be7-4d3c-8a68-1e85d4d515a0</RequestId>
</KMS>
```

`JSON` 格式

```
{
        "CiphertextBlob": "ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901qOjop4bTS****",
        "KeyId": "599fa825-17de-417e-9554-bb032cc6****",
        "KeyVersionId": "2ab1a983-7072-4bbc-a582-584b5bd8****",
        "Plaintext": "QmFzZTY0IGVuY29kZWQgcGxhaW50ZXh0",
        "RequestId": "7021b6ec-4be7-4d3c-8a68-1e85d4d515a0"
}
```

错误码

访问错误中心查看更多错误码。

8.19. GenerateDataKeyWithoutPlaintext

调用GenerateDataKeyWithoutPlaintext接口生成一个随机的数据密钥。您可以用数据密钥进行本地数据的加密。

此API随机生成一个数据密钥，并通过您指定的对称主密钥（Symmetric CMK）加密后，返回数据密钥的密文。此API和GenerateDataKey提供完全相同的功能，唯一的区别是此API不会返回数据密钥的明文。

您在请求中指定的CMK，仅用于数据密钥的加密，不会用于数据密钥的生成。KMS不会记录或存储随机生成的数据密钥。

说明

- 此API适用于不需要立即使用数据密钥完成数据加密的系统。系统需要加密时，通过调用Decrypt接口解开数据密钥的密文。
- 此API也适用于具有不同信任等级的分布式系统。例如：您的系统将数据按照既定划分策略存储到不同的分区中。其中的一个模块会预先创建不同的数据分区，对每一个分区分别产生不同的数据密钥。这一模块完成控制平面的初始化之后，并不参与数据的生产和消费，它是密钥分发者。而数据平面的模块，在产生和消费数据的时候，首先获取分区的数据密钥密文，在解开之后使用数据密钥的明文对数据执行加密或者解密操作，随后清除内存中的数据密钥明文。在这样的系统中，密钥分发者不需要获取到数据密钥的明文，只需要使用相关CMK的GenerateDataKeyWithoutPlaintext的权限；而数据的生产和消费者，不需要产生新的数据密钥，只需要使用相关CMK的Decrypt的权限。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GenerateDataKeyWithoutPlaintext	要执行的操作，取值：GenerateDataKeyWithoutPlaintext。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	主密钥（CMK）的全局唯一标识符。该参数也可以被指定为CMK绑定的别名，详情请参见别名使用说明。
KeySpec	String	否	AES_256	指定生成的数据密钥的长度，取值： • AES_256：256位的对称密钥 • AES_128：128位的对称密钥 说明 建议使用KeySpec或者NumberOfBytes来指定数据密钥长度。如果两者都不指定，KMS生成256位的数据密钥；如果两者都被指定，KMS会忽略KeySpec参数。
NumberOfBytes	Integer	否	256	指定生成的数据密钥的长度。 取值：1~1024。 单位：字节
EncryptionContext	Json	否	{"Example":"Example"}	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，详情请参见EncryptionContext说明。

返回数据

名称	类型	示例值	描述
CiphertextBlob	String	ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901qOjop4bTS****	数据密钥被指定CMK的主版本加密后的密文。
KeyId	String	599fa825-17de-417e-9554-bb032cc6****	CMK的全局唯一标识符。 说明 如果请求中的KeyId参数使用的是CMK的别名，在响应中会返回别名对应的CMK标志符。
KeyVersionId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	用于加密明文的密钥版本标志符。是指定CMK的主版本。
RequestId	String	7021b6ec-4be7-4d3c-8a68-1e85d4d515a0	本次请求的ID。

示例

请求示例

```
https://[Endpoint]/?Action=GenerateDataKeyWithoutPlaintext
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <RequestId>7021b6ec-4be7-4d3c-8a68-1e85d4d515a0</RequestId>
        <CiphertextBlob>ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901qOjop4bTS****</CiphertextBlob>
        <KeyId>599fa825-17de-417e-9554-bb032cc6****</KeyId>
        <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
</KMS>
```

`JSON` 格式

```
{
    "RequestId":"7021b6ec-4be7-4d3c-8a68-1e85d4d515a0",
    "CiphertextBlob":"ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901qOjop4bTS****",
    "KeyId":"599fa825-17de-417e-9554-bb032cc6****",
    "KeyVersionId":"2ab1a983-7072-4bbc-a582-584b5bd8****"
}
```

错误码

访问错误中心查看更多错误码。

8.20. ExportDataKey

调用ExportDataKey接口使用传入的公钥加密导出数据密钥。

调用GenerateDataKeyWithoutPlaintext获取主密钥（CMK）加密保护的数据密钥。当您需要将数据密钥分发到其它地域（Region）或者密码模块时，您可以调用ExportDataKey接口，返回指定公钥加密数据密钥的密文。

将公钥加密数据密钥的密文，导入到公钥对应私钥所在的密码模块，可实现KMS到密码模块的密钥分发，保障了数据密钥分发过程的安全性。该密钥导入到密码模块后，可用于实现相应数据的加解密运算。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ExportDataKey	要执行的操作，取值：ExportDataKey。
CiphertextBlob	String	是	ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901q********	主密钥（CMK）加密的数据密钥的密文。
PublicKeyBlob	String	是	MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAndKfC2ReLL2+y8a0+ZBBeAft/uBYo86GZiYJuflqgUzKxpyuvlo3uQkBv6b+nx+0tz8g8v7GhpPWMSW5L9mNHYsvYFsa7jTxsYdt17yj6GlUHPuMIs8hr5qbwl38IHU1ila7nYWwE2fb3ePOvLDACRJVgGpU0yxioW80d2QD+9aU4jF5dlAahcfgsNzo2CXzCUc1+xbmNuq7Rp+H9VJB9dyYOwqnW3RhOLBo21FzpORapf0UiRlrHRpk1V6ez+aE1dofaYh/9bh0m6iox j7j5hpZbWccuEZTMBKd+cbuBkRhJzc6Tti6qwZbDiu4fUwbZS0Tqpuo1UadiyxMW**********	Base64格式的公钥。
WrappingAlgorithm	String	是	RSAES_OAEP_SHA_256	使用PublicKeyBlob所指定的公钥，加密（Wrap）数据密钥时的加密算法。算法详情，请参见AsymmetricDecrypt。 取值： • RSAES_OAEP_SHA_256 • RSAES_OAEP_SHA_1 • SM2PKE
WrappingKeySpec	String	是	RSA_2048	PublicKeyBlob的密钥类型。密钥类型详情，请参见非对称密钥简介。 取值： • RSA_2048 • EC_SM2
EncryptionContext	Json	否	{"Example":"Example"}	key/value的JSON字符串。EncryptionContext是使用CMK加密数据密钥时传入的加密上下文，详情请参见EncryptionContext说明。

返回数据

名称	类型	示例值	描述
ExportedDataKey	String	BQKP+1zK6+ZEMxTP5qaVzcsgXtWplYBKm0NXdSnB5FzliFxE1bSiu4dnEllca2JpeH7yz1/S6fed630H+hlH6DoM25fTLNcKj+mFB0Xnh9m2+HN59Mn4qyTfcUeadnfCXSWcGBouhXFwcdd2rJ3n337bzTf4jm659gZu3L0i6PLuxM9p7mqdwO0cKJPfGVfhnfMz+f4alMg79WB/NNyE2lyX7/qxvV49ObNrrJbKSFiz8Djocaf0IESNLMbfYI5bXjWkJlX92DQbKhibtQW8ZOJ//ZC6t0AWcUoKL6QDm/dg5koQalcleRinpB+QadFm894sLbVZ9+N4GVs*******	公钥加密保护导出的数据密钥。

名称	类型	示例值	描述
KeyId	String	202b9877-5a25-46e3-a763-e20791b5****	解密传入的数据密钥密文使用的主密钥ID。 主密钥的全局唯一标识符。
KeyVersionId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	用于解密传入的数据密钥密文的密钥版本标识符。
RequestId	String	4bd560a1-729e-45f1-a3d9-b2a33d61046b	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ExportDataKey
&CiphertextBlob=ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901q*
*******
&PublicKeyBlob=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAndKfC2ReLL2+y8a0+ZBBeAft/uBYo86GZiYJuflqgUzKxpyuvlo3uQkBv6b+nx+0tz8g8v7GhpPWMSW5L9mNHYsvYF
sa7jTxsYdt17yj6GlUHPuMIs8hr5qbwl38IHU1iIa7nYWwE2fb3ePOvLDACRJVgGpU0yxioW80d2QD+9aU4jF5dlAahcfgsNzo2CXzCUc1+xbmNuq7Rp+H9VJB9dyYOwqnW3RhOLBo21FzpORapf0
UiRlrHRpk1V6ez+aE1dofaYh/9bh0m6ioxj7j5hpZbWccuEZTMBKd+cbuBkRhJzc6Tti6qwZbDiu4fUwbZS0TqpuolUadiyxMW********
&WrappingAlgorithm=RSAES_OAEP_SHA_256
&WrappingKeySpec=RSA_2048
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <KeyId>202b9877-5a25-46e3-a763-e20791b5****</KeyId>
        <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
        <ExportedDataKey>BQKP+1zK6+ZEMxTP5qaVzcsgXtWplYBKm0NXdSnB5FzliFxE1bSiu4dnEIlca2JpeH7yz1/S6fed630H+hIH6DoM25fTLNcKj+mFB0Xnh9m2+HN59Mn4qyTfcU
eadnfCXSWcGBouhXFwcdd2rJ3n337bzTf4jm659gZu3L0i6PLuxM9p7mqdwO0cKJPfGVfhnfMz+f4alMg79WB/NNyE2lyX7/qxvV49ObNrrJbKSFiz8Djocaf0IESNLMbfYI5bXjWkJlX92DQbKhi
btQW8ZOJ//ZC6t0AWcUoKL6QDm/dg5koQalcleRinpB+QadFm894sLbVZ9+N4GVs*******</ExportedDataKey>
        <RequestId>4bd560a1-729e-45f1-a3d9-b2a33d61046b</RequestId>
</KMS>
```

`JSON` 格式

```
{
    "KeyId": "202b9877-5a25-46e3-a763-e20791b5****",
    "KeyVersionId": "2ab1a983-7072-4bbc-a582-584b5bd8****",
    "ExportedDataKey": "BQKP+1zK6+ZEMxTP5qaVzcsgXtWplYBKm0NXdSnB5FzliFxE1bSiu4dnEIlca2JpeH7yz1/S6fed630H+hIH6DoM25fTLNcKj+mFB0Xnh9m2+HN59Mn4qyTfcUead
nfCXSWcGBouhXFwcdd2rJ3n337bzTf4jm659gZu3L0i6PLuxM9p7mqdwO0cKJPfGVfhnfMz+f4alMg79WB/NNyE2lyX7/qxvV49ObNrrJbKSFiz8Djocaf0IESNLMbfYI5bXjWkJlX92DQbKhibtQ
W8ZOJ//ZC6t0AWcUoKL6QDm/dg5koQalcleRinpB+QadFm894sLbVZ9+N4GVs*******",
    "RequestId": "4bd560a1-729e-45f1-a3d9-b2a33d61046b"
}
```

错误码

HttpCode	错误码	错误信息	描述
500	InternalFailure	Internal Failure.	内部错误。建议重试，如果多次重试报错请提交工单。

访问错误中心查看更多错误码。

8.21. GenerateAndExportDataKey

调用GenerateAndExportDataKey接口随机生成一个数据密钥，通过您指定的主密钥（CMK）和公钥加密后，返回CMK加密数据密钥的密文和公钥加密数据密钥的密文。

建议您使用以下方式将数据密钥导入到密码模块中，用于数据加密和数据解密：

- 调用GenerateAndExportDataKey接口，获得CMK加密和指定公钥加密的数据密钥。
- 将CMK加密数据密钥得到的密文保存在KMS凭据管家，或者云数据库等存储服务中，用于密钥的备份和恢复。
- 将公钥加密数据密钥的密文，导入到公钥对应私钥所在的密码模块，实现KMS到密码模块的密钥分发，使用数据密钥对相应的数据进行加解密运算。

> **说明** 在请求中指定的CMK，仅会被用作数据密钥的加密，和数据密钥的生成没有关系。KMS也不会记录或存储随机生成的数据密钥，您需要负责记录数据密钥或数据密钥的密文。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GenerateAndExportDataKey	要执行的操作，取值：GenerateAndExportDataKey。

名称	类型	是否必选	示例值	描述
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	主密钥（CMK）的全局唯一标识符。该参数也可以被指定为CMK绑定的别名，详情请参见别名使用说明。
PublicKeyBlob	String	是	MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAndKfC2ReLL2+y8a0+ZBBeAft/uBYo86GZiYJuflqgUzKxpyuvlo3uQkBv6b+nx+0tz8g8v7GhpPWMSW5L9mNHYsvYFsa7jTxsYdt17yj6GlUHPuMIs8hr5qbwl38lHU1ila7nYWwE2fb3ePOvLDACRJVgGpU0yxioW80d2QD+9aU4jF5dlAahcfgsNzo2CXzCUc1+xbmNuq7Rp+H9VJB9dyYOwqnW3RhOLBo21FzpORapf0UiRlrHRpk1V6ez+aE1dofaYh/9bh0m6iox j7j5hpZbWccuEZTMBKd+cbuBkRhJzc6Tti6qwZbDiu4fUwbZS0Tqpuo1UadiyxMW**********	Base64编码的公钥。
WrappingAlgorithm	String	是	RSAES_OAEP_SHA_256	使用PublicKeyBlob所指定的公钥，加密（Wrap）数据密钥时的加密算法。算法详情，请参见AsymmetricDecrypt。 取值： • RSAES_OAEP_SHA_256 • RSAES_OAEP_SHA_1 • SM2PKE
WrappingKeySpec	String	是	RSA_2048	PublicKeyBlob密钥的类型。密钥类型详情，请参见非对称密钥简介。 取值： • RSA_2048 • EC_SM2
KeySpec	String	否	AES_256	指定生成的数据密钥的长度，取值： • AES_256：256位的对称密钥。 • AES_128：128位的对称密钥。 说明 建议使用KeySpec或者NumberOfBytes来指定数据密钥长度。如果两者都不指定，KMS生成256位的数据密钥；如果两者都被指定，KMS会忽略KeySpec参数。
NumberOfBytes	Integer	否	32	指定生成的数据密钥的长度。 取值：1~1024。 单位：字节。
EncryptionContext	Json	否	{"Example":"Example"}	key/value对的JSON字符串，如果指定了该参数，则在解密或者使用其他密钥转加密时需要提供同样的参数，详情请参见EncryptionContext说明。

返回数据

名称	类型	示例值	描述
CiphertextBlob	String	ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901qOjop4bTS****	数据密钥被指定CMK的主版本加密后的密文。
ExportedDataKey	String	BQKP+1zK6+ZEMxTP5qaVzcsgXtWplYBKm0NXdSnB5FzliFxE1bSiu4dnEllca2JpeH7yz1/S6fed630H+hIH6DoM25fTLNcKj+mFB0Xnh9m2+HN59Mn4qyTfcUeadnfCXSWcGBouhXFwcdd2rJ3n337bzTf4jm659gZu3L0i6PLuxM9p7mqdwO0cKJPfGVfhnfMz+f4alMg79WB/NNyE2lyX7/qxvV49ObNrrJbKSFiz8Djocaf0IESNLMbfYI5bXjWkJlX92DQbKhibtQW8ZOJ//ZC6t0AWcUoKL6QDm/dg5koQalcleRinpB+QadFm894sLbVZ9+N4GVs*******	公钥加密保护导出的数据密钥。

名称	类型	示例值	描述
KeyId	String	599fa825-17de-417e-9554-bb032cc6****	CMK的全局唯一标识符。 说明 如果请求中的KeyId参数使用的是CMK的别名，在响应中会返回别名对应的CMK标识符。
KeyVersionId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	用于加密明文的密钥版本标识符。是指定CMK的主版本。
RequestId	String	7021b6ec-4be7-4d3c-8a68-1e85d4d515a0	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GenerateAndExportDataKey
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&PublicKeyBlob=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAndKfC2ReLL2+y8a0+ZBBeAft/uBYo86GZiYJuflqgUzKxpyuvlo3uQkBv6b+nx+0tz8g8v7GhpPWMSW5L9mNHYsvYFsa7jTxsYdt17yj6GlUHPuMIs8hr5qbwl38IHU1iIa7nYWwE2fb3ePOvLDACRJVgGpU0yxioW80d2QD+9aU4jF5dlAahcfgsNzo2CXzCUc1+xbmNuq7Rp+H9VJB9dyYOwqnW3RhOLBo21FzpORapf0UiRlrHRpk1V6ez+aE1dofaYh/9bh0m6ioxj7j5hpZbWccuEZTMBKd+cbuBkRhJzc6Tti6qwZbDiu4fUwbZS0TqpuolUadiyxMW********
&WrappingAlgorithm=RSAES_OAEP_SHA_256
&WrappingKeySpec=RSA_2048
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <KeyId>202b9877-5a25-46e3-a763-e20791b5****</KeyId>
        <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
        <CiphertextBlob>ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901qOjop4bTS****</CiphertextBlob>
        <ExportedDataKey>BQKP+1zK6+ZEMxTP5qaVzcsgXtWplYBKm0NXdSnB5FzliFxE1bSiu4dnEIlca2JpeH7yz1/S6fed630H+hIH6DoM25fTLNcKj+mFB0Xnh9m2+HN59Mn4qyTfcUeadnfCXSWcGBouhXFwcdd2rJ3n337bzTf4jm659gZu3L0i6PLuxM9p7mqdwO0cKJPfGVfhnfMz+f4alMg79WB/NNyE2lyX7/qxvV49ObNrrJbKSFiz8Djocaf0IESNLMbfYI5bXjWkJlX92DQbKhibtQW8ZOJ//ZC6t0AWcUoKL6QDm/dg5koQalcleRinpB+QadFm894sLbVZ9+N4GVs*******</ExportedDataKey>
        <RequestId>4bd560a1-729e-45f1-a3d9-b2a33d61046b</RequestId>
</KMS>
```

`JSON` 格式

```
{
    "KeyId": "202b9877-5a25-46e3-a763-e20791b5****",
    "KeyVersionId": "2ab1a983-7072-4bbc-a582-584b5bd8****",
    "CiphertextBlob":"ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901qOjop4bTS****",
    "ExportedDataKey": "BQKP+1zK6+ZEMxTP5qaVzcsgXtWplYBKm0NXdSnB5FzliFxE1bSiu4dnEIlca2JpeH7yz1/S6fed630H+hIH6DoM25fTLNcKj+mFB0Xnh9m2+HN59Mn4qyTfcUeadnfCXSWcGBouhXFwcdd2rJ3n337bzTf4jm659gZu3L0i6PLuxM9p7mqdwO0cKJPfGVfhnfMz+f4alMg79WB/NNyE2lyX7/qxvV49ObNrrJbKSFiz8Djocaf0IESNLMbfYI5bXjWkJlX92DQbKhibtQW8ZOJ//ZC6t0AWcUoKL6QDm/dg5koQalcleRinpB+QadFm894sLbVZ9+N4GVs*******",
    "RequestId": "207596a2-36d3-4840-b1bd-f87044699bd7"
}
```

错误码

HttpCode	错误码	错误信息	描述
404	Forbidden.KeyNotFound	The specified Key is not found.	指定的密钥不存在。

访问错误中心查看更多错误码。

8.22. Decrypt

调用Decrypt接口解密CiphertextBlob中的密文。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	Decrypt	要执行的操作，取值：Decrypt。

名称	类型	是否必选	示例值	描述
CiphertextBlob	String	是	DZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmaaSl+TztSIMe43nbTH/Z1Wr4XfLftKhAciUmDQXuMRl4WTvKhxjMThjK****	待解密的密文。 密文可以通过以下API生成： • GenerateDataKey • Encrypt • GenerateDataKeyWithoutPlaintext
EncryptionContext	Json	否	{"Example":"Example"}	key/value的JSON字符串。 说明 如果在调用GenerateDataKey、Encrypt或GenerateDataKeyWithoutPlaintext加密时指定了EncryptionContext，则需要在解密时提供同样的参数。更多信息，请参见EncryptionContext说明。

返回数据

名称	类型	示例值	描述
KeyId	String	202b9877-5a25-46e3-a763-e20791b5****	解密密文使用的主密钥ID。 主密钥的全局唯一标识符。
KeyVersionId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	主密钥下用于解密密文的密钥版本标识符。
Plaintext	String	tRYXuCwgja12xxO1N/gZERDDCLw9doZEQiPDk/Bv****	解密后的明文。
RequestId	String	207596a2-36d3-4840-b1bd-f87044699bd7	请求ID。

示例

请求示例

```
https://[Endpoint]/?Action=Decrypt
&CiphertextBlob=DZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmaaSl+TztSIMe43nbTH/Z1Wr4XfLftKhAciUmDQXuMRl4WTvKhxjMThjK****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <KeyId>202b9877-5a25-46e3-a763-e20791b5****</KeyId>
        <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
        <Plaintext>tRYXuCwgja12xxO1N/gZERDDCLw9doZEQiPDk/Bv****</Plaintext>
        <RequestId>4bd560a1-729e-45f1-a3d9-b2a33d61046b</RequestId>
</KMS>
```

`JSON` 格式

```
{
    "KeyId": "202b9877-5a25-46e3-a763-e20791b5****",
    "KeyVersionId": "2ab1a983-7072-4bbc-a582-584b5bd8****",
    "Plaintext": "tRYXuCwgja12xxO1N/gZERDDCLw9doZEQiPDk/Bv****",
    "RequestId": "207596a2-36d3-4840-b1bd-f87044699bd7"
}
```

错误码

访问错误中心查看更多错误码。

8.23. ReEncrypt

调用ReEncrypt接口对密文进行转加密。即先解密密文，然后将解密得到的数据或者数据密钥使用新的主密钥再次进行加密，返回加密结果。

ReEncrypt使用场景如下：

- 主密钥（CMK）进行轮转后，使用轮转后最新的密钥版本对数据进行重新加密。自动轮转密钥详情，请参见自动轮转密钥。
- 主密钥不变，改变加密上下文的内容，进行重新加密。
- 将主密钥加密的数据或者数据密钥在KMS内部使用其它的主密钥进行重新加密。

ReEncrypt权限设置如下：

- 需要有操作源主密钥的kms:ReEncryptFrom权限。
- 需要有操作目的主密钥的kms:ReEncryptTo权限。
- 可以设置kms:ReEncrypt*用于表示上述两个操作的权限。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ReEncrypt	要执行的操作，取值：ReEncrypt。
CiphertextBlob	String	是	ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901q********	待转加密的密文。 该参数可以为对称加密或非对称加密返回的密文数据。 • 对称加密：调用Encrypt、GenerateDataKey、GenerateDataKeyWithoutPlaintext或GenerateAndExportDataKey接口返回的密文数据。 • 非对称加密：可以是调用GenerateAndExportDataKey接口返回的公钥加密数据，也可以是外部系统使用非对称公钥加密的数据。
DestinationKeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	对密文解密后再次加密时使用的对称主密钥ID。
SourceKeyId	String	否	5c438b18-05be-40ad-b6c2-3be6752c****	解密密文时使用的主密钥ID。 主密钥的全局唯一标识符。 说明 当CiphertextBlob是非对称加密返回的公钥加密数据时需要指定该参数。
SourceKeyVersionId	String	否	2ab1a983-7072-4bbc-a582-584b5bd8****	用于解密密文的密钥版本标识符。 说明 当CiphertextBlob是非对称加密返回的公钥加密数据时需要指定该参数。
SourceEncryptionAlgorithm	String	否	RSAES_OAEP_SHA_256	CiphertextBlob是公钥加密结果时，指定公钥加密的算法。算法详情，请参见AsymmetricDecrypt。 取值： • RSAES_OAEP_SHA_256 • RSAES_OAEP_SHA_1 • SM2PKE 说明 当CiphertextBlob是非对称加密返回的公钥加密数据时需要指定该参数。
SourceEncryptionContext	Json	否	{"Example":"Example"}	key/value的JSON字符串。如果在Encrypt、GenerateDataKey、GenerateDataKeyWithoutPlaintext或GenerateAndExportDataKey API中指定了该参数，则需要提供同样的参数才能解密，详情请参见EncryptionContext说明。 说明 当CiphertextBlob是对称加密返回的密文数据时需要指定该参数。
DestinationEncryptionContext	Json	否	{"Example":"Example"}	key/value的JSON字符串，用于目标主密钥加密时的加密上下文。

返回数据

名称	类型	示例值	描述
CiphertextBlob	String	DZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmaaSl+TztSlMe43nbTH/Z1Wr4XfLftKhAciUmDQXuMRl4WTvKhxjMThjK****	使用指定的主密钥进行再次加密得到的密文。
KeyId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	解密密文使用的主密钥ID。 主密钥的全局唯一标识符。
KeyVersionId	String	202b9877-5a25-46e3-a763-e20791b5****	主密钥下用于解密密文的密钥版本标识符。
RequestId	String	207596a2-36d3-4840-b1bd-f87044699bd7	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ReEncrypt
&CiphertextBlob=ODZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmS7FmDBBQ0BkKsQrtRnidtPwirmDcS0ZuJCU41xxAAWk4Z8qsADfbV0b+i6kQmlvj79dJdGOvtX69Uycs901q*
*******
&DestinationKeyId=1234abcd-12ab-34cd-56ef-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <KeyId>202b9877-5a25-46e3-a763-e20791b5****</KeyId>
        <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
        <CiphertextBlob>DZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmaaSl+TztSIMe43nbTH/Z1Wr4XfLftKhAciUmDQXuMRl4WTvKhxjMThjK****</CiphertextBlob
>
        <RequestId>4bd560a1-729e-45f1-a3d9-b2a33d61046b</RequestId>
</KMS>
```

`JSON` 格式

```
{
    "KeyId": "202b9877-5a25-46e3-a763-e20791b5****",
    "KeyVersionId": "2ab1a983-7072-4bbc-a582-584b5bd8****",
    "CiphertextBlob": "DZhOWVmZDktM2QxNi00ODk0LWJkNGYtMWZjNDNmM2YyYWJmaaSl+TztSIMe43nbTH/Z1Wr4XfLftKhAciUmDQXuMRl4WTvKhxjMThjK****",
    "RequestId": "4bd560a1-729e-45f1-a3d9-b2a33d61046b"
}
```

错误码

HttpCode	错误码	错误信息	描述
500	InternalFailure	Internal Failure.	内部错误。建议重试，如果多次重试报错请提交工单。

访问错误中心查看更多错误码。

8.24. AsymmetricSign

调用AsymmetricSign接口使用非对称密钥进行签名。

仅支持**Usage**为**SIGN/VERIFY**的非对称密钥。支持的签名算法如下表：

KeySpec	Algorithm	说明
RSA_2048	RSA_PSS_SHA_256	RSASSA-PSS using SHA-256 and MGF1 with SHA-256
RSA_2048	RSA_PKCS1_SHA_256	RSASSA-PKCS1-v1_5 using SHA-256
RSA_3072	RSA_PSS_SHA_256	RSASSA-PSS using SHA-256 and MGF1 with SHA-256
RSA_3072	RSA_PKCS1_SHA_256	RSASSA-PKCS1-v1_5 using SHA-256
EC_P256	ECDSA_SHA_256	ECDSA on the P-256 Curve(secp256r1) with a SHA-256 digest
EC_P256K	ECDSA_SHA_256	ECDSA on the P-256K Curve(secp256k1) with a SHA-256 digest
EC_SM2	SM2DSA	SM2椭圆曲线数字签名算法

说明 按照国家标准GBT 32918，计算SM2签名值时，**Digest**参数不是对原始消息直接计算SM3摘要，而是对Z(A)和M的拼接值计算的摘要，其中M是待签名的原始消息，Z(A)是GBT 32918中定义的用户A的杂凑值。

本文将提供一个示例，使用密钥ID为 `5c438b18-05be-40ad-b6c2-3be6752c****` 、密钥版本ID为 `2ab1a983-7072-4bbc-a582-584b5bd8****` 的非对称密钥，通过签名算法 `RSA_PSS_SHA_256` 对摘要信息 `ZOyIygCyaOW6GjVnihtTFtIS9PNmskdyMlNKiuy****=` 进行签名。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述

名称	类型	是否必选	示例值	描述
Action	String	是	AsymmetricSign	要执行的操作，取值：AsymmetricSign。
Algorithm	String	是	RSA_PSS_SHA_256	签名算法。
Digest	String	是	ZOyIygCyaOW6GjVnihtTFtIS9PNmskdyMlNKiu****=	使用Algorithm中对应的哈希算法，对原始消息生成的摘要。 说明 • 使用Base64编码。 • 关于如何计算消息摘要，请参见非对称数字签名的签名预处理：计算消息摘要章节。
KeyId	String	是	5c438b18-05be-40ad-b6c2-3be6752c****	主密钥（CMK）的全局唯一标识符。 说明 该参数也可以被指定为主密钥绑定的别名。更多信息，请参见别名使用说明。
KeyVersionId	String	是	2ab1a983-7072-4bbc-a582-584b5bd8****	密钥版本ID。密钥版本的全局唯一标识符。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
KeyId	String	5c438b18-05be-40ad-b6c2-3be6752c****	主密钥的全局唯一标识符。 说明 如果请求中的KeyId参数使用的是主密钥的别名，在响应中会返回别名对应的主密钥标识符。
KeyVersionId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	密钥版本ID。密钥版本的全局唯一标识符。
Value	String	M2CceNZH00ZgL9ED/ZHFp21YRAvYeZHknJUc207OCZ0N9wNn9As4z2bON3FF3je+1Nu+2+/8Zj50HpMTpzYpMp2R93cYmACCmhaYoKydxylbyGzJR8y9likZRCrkD38lRoS40aBBvv/6iRKzQuo9EGYVcel36cMNg00VmYNBy3pa1rwg3gA4l3cy6kjayZja1WGPkVhrVKsrJMdbpl0ApLjXKuD8rw1n1XLCwCUEL5eLPljTZaAveqdOFQOiZnZEGI27qIiZe7I1fN8tcz6anS/gTM7xRKE++5egEvRWlTQQTJeApnPSiUPA+8ZykNdelQsOQh5SrGoyI4A5pq****==	计算出来的签名。 说明 使用Base64编码。
RequestId	String	475f1620-b9d3-4d35-b5c6-3fbdd941423d	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=AsymmetricSign
&Algorithm=RSA_PSS_SHA_256
&Digest=ZOyIygCyaOW6GjVnihtTFtIS9PNmskdyMlNKiu****=
&KeyId=5c438b18-05be-40ad-b6c2-3be6752c****
&KeyVersionId=2ab1a983-7072-4bbc-a582-584b5bd8****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
    <KeyId>5c438b18-05be-40ad-b6c2-3be6752c****</KeyId>
    <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
    <Value>M2CceNZH00ZgL9ED/ZHFp21YRAvYeZHknJUc207OCZ0N9wNn9As4z2bON3FF3je+1Nu+2+/8Zj50HpMTpzYpMp2R93cYmACCmhaYoKydxylbyGzJR8y9likZRCrkD38lRoS40aBBvv
/6iRKzQuo9EGYVcel36cMNg00VmYNBy3pa1rwg3gA4l3cy6kjayZja1WGPkVhrVKsrJMdbpl0ApLjXKuD8rw1n1XLCwCUEL5eLPljTZaAveqdOFQOiZnZEGI27qIiZe7I1fN8tcz6anS/gTM7xRKE
++5egEvRWlTQQTJeApnPSiUPA+8ZykNdelQsOQh5SrGoyI4A5pq****==</Value>
    <RequestId>475f1620-b9d3-4d35-b5c6-3fbdd941423d</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "KeyId": "5c438b18-05be-40ad-b6c2-3be6752c****",
  "KeyVersionId": "2ab1a983-7072-4bbc-a582-584b5bd8****",
  "Value": "M2CceNZH00ZgL9ED/ZHFp21YRAvYeZHknJUc207OCZ0N9wNn9As4z2bON3FF3je+1Nu+2+/8Zj50HpMTpzYpMp2R93cYmACCmhaYoKydxylbyGzJR8y9likZRCrkD38lRoS40aBBvv/6iRKzQuo9EGYVcel36cMNg00VmYNBy3pa1rwg3gA4l3cy6kjayZja1WGPkVhrVKsrJMdbpl0ApLjXKuD8rw1n1XLCwCUEL5eLPljTZaAveqdOFQOiZnZEGI27qIiZe7I1fN8tcz6anS/gTM7xRKE++5egEvRWlTQQTJeApnPSiUPA+8ZykNdelQsOQh5SrGoyI4A5pq****==",
  "RequestId": "475f1620-b9d3-4d35-b5c6-3fbdd941423d"
}
```


错误码

访问错误中心查看更多错误码。

8.25. AsymmetricVerify

调用AsymmetricVerify接口使用非对称密钥进行验签。

仅支持**Usage**为**SIGN/VERIFY**的非对称密钥。支持的签名算法如下表：

KeySpec	Algorithm	说明
RSA_2048	RSA_PSS_SHA_256	RSASSA-PSS using SHA-256 and MGF1 with SHA-256
RSA_2048	RSA_PKCS1_SHA_256	RSASSA-PKCS1-v1_5 using SHA-256
RSA_3072	RSA_PSS_SHA_256	RSASSA-PSS using SHA-256 and MGF1 with SHA-256
RSA_3072	RSA_PKCS1_SHA_256	RSASSA-PKCS1-v1_5 using SHA-256
EC_P256	ECDSA_SHA_256	ECDSA on the P-256 Curve(secp256r1) with a SHA-256 digest
EC_P256K	ECDSA_SHA_256	ECDSA on the P-256K Curve(secp256k1) with a SHA-256 digest
EC_SM2	SM2DSA	SM2椭圆曲线数字签名算法

> **说明** 按照国家标准GBT 32918，计算SM2签名值时，**Digest**参数不是对原始消息直接计算SM3摘要，而是对Z(A)和M的拼接值计算的摘要，其中M是待签名的原始消息，Z(A)是GBT 32918中定义的用户A的杂凑值。

本文将提供一个示例，使用密钥ID为`5c438b18-05be-40ad-b6c2-3be6752c****`、密钥版本ID为`2ab1a983-7072-4bbc-a582-584b5bd8****`的非对称密钥，通过签名算法RSA_PSS_SHA_256对摘要信息`ZOyIygCyaOW6GjVnihtTFtIS9PNmskdyMlNKiuyjfzw=`生成的签名值`M2CceNZH00ZgL9ED/ZHFp21YRAvYeZHknJUc207OCZ0N9wNn9As4z2bON3FF3je+1Nu+2+/8Zj50HpMTpzYpMp2R93cYmACCmhaYoKydxylbyGzJR8y9likZRCrkD38lRoS40aBBvv/6iRKzQuo9EG`行验证。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	AsymmetricVerify	要执行的操作，取值：AsymmetricVerify。
Algorithm	String	是	RSA_PSS_SHA_256	签名算法。
Digest	String	是	ZOyIygCyaOW6GjVnihtTFtIS9PNmskdyMlNKiuy****=	使用**Algorithm**中对应的哈希算法，对原始消息生成的摘要。 > **说明** 使用Base64编码。
KeyId	String	是	5c438b18-05be-40ad-b6c2-3be6752c****	主密钥（CMK）的全局唯一标识符。 > **说明** 该参数也可以被指定为主密钥绑定的别名。更多信息，请参加见别名使用说明。
KeyVersionId	String	是	2ab1a983-7072-4bbc-a582-584b5bd8****	密钥版本ID。密钥版本的全局唯一标识符。

名称	类型	是否必选	示例值	描述
Value	String	是	M2CceNZH00ZgL9ED/ZHFp21YRAvYeZHknJUc207OCZ0N9wNn9As4z2bON3FF3je+1Nu+2+/8Zj50HpMTpzYpMp2R93cYmACCmhaYoKydxylbyGzJR8y9likZRCrkD38lRoS40aBBvv/6iRKzQuo9EGYVcel36cMNg00VmYNBy3pa1rwg3gA4l3cy6kjayZja1WGPkVhrVKsrJMdbpl0ApLjXKuD8rw1n1XLCwCUEL5eLPljTZaAveqdOFQOiZnZEGI27qIiZe7I1fN8tcz6anS/gTM7xRKE++5egEvRWlTQQTJeApnPSiUPA+8ZykNdelQsOQh5SrGoyI4A5pq****==	待验证的签名值。 说明 使用Base64编码。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
KeyId	String	5c438b18-05be-40ad-b6c2-3be6752c****	主密钥的全局唯一标识符。 说明 如果请求中的KeyId参数使用的是主密钥的别名，在响应中会返回别名对应的主密钥标识符。
KeyVersionId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	对明文数据进行加密的主密钥版本号。
Value	Boolean	true	签名验证是否通过。
RequestId	String	475f1620-b9d3-4d35-b5c6-3fbdd941423d	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=AsymmetricVerify
&Algorithm=RSA_PSS_SHA_256
&Digest=ZOyIygCyaOW6GjVnihtTFtIS9PNmskdyMlNKiuy****=
&KeyId=5c438b18-05be-40ad-b6c2-3be6752c****
&KeyVersionId=2ab1a983-7072-4bbc-a582-584b5bd8****
&Value=M2CceNZH00ZgL9ED/ZHFp21YRAvYeZHknJUc207OCZ0N9wNn9As4z2bON3FF3je+1Nu+2+/8Zj50HpMTpzYpMp2R93cYmACCmhaYoKydxylbyGzJR8y9likZRCrkD38lRoS40aBBvv/6iRKzQuo9EGYVcel36cMNg00VmYNBy3pa1rwg3gA4l3cy6kjayZja1WGPkVhrVKsrJMdbpl0ApLjXKuD8rw1n1XLCwCUEL5eLPljTZaAveqdOFQOiZnZEGI27qIiZe7I1fN8tcz6anS/gTM7xRKE++5egEvRWlTQQTJeApnPSiUPA+8ZykNdelQsOQh5SrGoyI4A5pq****==
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
    <KeyId>5c438b18-05be-40ad-b6c2-3be6752c****</KeyId>
    <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
    <Value>true</Value>
    <RequestId>475f1620-b9d3-4d35-b5c6-3fbdd941423d</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "KeyId": "5c438b18-05be-40ad-b6c2-3be6752c****",
  "KeyVersionId": "2ab1a983-7072-4bbc-a582-584b5bd8****",
  "Value": true,
  "RequestId": "475f1620-b9d3-4d35-b5c6-3fbdd941423d"
}
```

错误码

访问错误中心查看更多错误码。

8.26. AsymmetricDecrypt

调用AsymmetricDecrypt接口使用非对称密钥进行解密。

仅支持**Usage**为**ENCRYPT/DECRYPT**的非对称密钥。支持的加密算法如下表：

KeySpec	Algorithm	说明	密文长度（字节）
RSA_2048	RSAES_OAEP_SHA_256	RSAES-OAEP using SHA-256 and MGF1 with SHA-256	256
RSA_2048	RSAES_OAEP_SHA_1	RSAES-OAEP using SHA1 and MGF1 with SHA1	256
RSA_3072	RSAES_OAEP_SHA_256	RSAES-OAEP using SHA-256 and MGF1 with SHA-256	384
RSA_3072	RSAES_OAEP_SHA_1	RSAES-OAEP using SHA1 and MGF1 with SHA1	384
EC_SM2	SM2PKE	SM2椭圆曲线公钥加密算法	最大6144

本文将提供一个示例，使用密钥ID为 `5c438b18-05be-40ad-b6c2-3be6752c****` 、密钥版本ID为 `2ab1a983-7072-4bbc-a582-584b5bd8****` 的非对称密钥，通过解密算法 `RSAES_OAEP_SHA_1` 对密文 `BQKP+1zK6+ZEMxTP5qaVzcsgXtWplYBKm0NXdSnB5FzliFxE1bSiu4dnEIlca2JpeH7yz1/S6fed630H+hIH6DoM25fTLNcKj+mFB0Xnh9m2+HN59Mn4qyTfcUeadnfCXSWcGBouhXFwcdd2rJ3n33` 行解密。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	AsymmetricDecrypt	要执行的操作，取值：AsymmetricDecrypt。
KeyId	String	是	5c438b18-05be-40ad-b6c2-3be6752c****	主密钥（CMK）的全局唯一标识符。 说明 该参数也可以被指定为CMK绑定的别名。具体操作，请参见别名使用说明。
Algorithm	String	是	RSAES_OAEP_SHA_1	解密算法。
CiphertextBlob	String	是	BQKP+1zK6+ZEMxTP5qaVzcsgXtWplYBKm0NXdSnB5FzliFxE1bSiu4dnEIlca2JpeH7yz1/S6fed630H+hIH6DoM25fTLNcKj+mFB0Xnh9m2+HN59Mn4qyTfcUeadnfCXSWcGBouhXFwcdd2rJ3n337bzTf4jm659gZu3L0i6PLuxM9p7mqdwO0cKJPfGVfhnfMz+f4alMg79WB/NNyE2lyX7/qxvV49ObNrrJbKSFiz8Djocaf0IESNLMbfYI5bXjWkJlX92DQbKhibtQW8ZOJ//ZC6t0AWcUoKL6QDm/dg5koQalcleRinpB+QadFm894sLbVZ9+N4GVsv1W****==	解密密文。 说明 • 使用Base64编码。 • 您可以调用AsymmetricEncrypt接口生成密文。
KeyVersionId	String	是	2ab1a983-7072-4bbc-a582-584b5bd8****	密钥版本ID。密钥版本的全局唯一标识符。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
KeyId	String	5c438b18-05be-40ad-b6c2-3be6752c****	主密钥的全局唯一标识符。 说明 如果请求中的KeyId参数使用的是主密钥的别名，在响应中会返回别名对应的主密钥标识符。
KeyVersionId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	对明文数据进行加密的主密钥版本号。
Plaintext	String	SGVsbG8gd29ybGQ=	解密后的明文，使用Base64编码。
RequestId	String	475f1620-b9d3-4d35-b5c6-3fbdd941423d	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=AsymmetricDecrypt
&KeyId=5c438b18-05be-40ad-b6c2-3be6752c****
&Algorithm=RSAES_OAEP_SHA_1
&CiphertextBlob=BQKP+1zK6+ZEMxTP5qaVzcsgXtWplYBKm0NXdSnB5FzliFxE1bSiu4dnEIlca2JpeH7yz1/S6fed630H+hIH6DoM25fTLNcKj+mFB0Xnh9m2+HN59Mn4qyTfcUeadnfCXSWcG
BouhXFwcdd2rJ3n337bzTf4jm659gZu3L0i6PLuxM9p7mqdwO0cKJPfGVfhnfMz+f4alMg79WB/NNyE2lyX7/qxvV49ObNrrJbKSFiz8Djocaf0IESNLMbfYI5bXjWkJlX92DQbKhibtQW8ZOJ//Z
C6t0AWcUoKL6QDm/dg5koQalcleRinpB+QadFm894sLbVZ9+N4GVsv1W****==
&KeyVersionId=2ab1a983-7072-4bbc-a582-584b5bd8****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
    <KeyId>5c438b18-05be-40ad-b6c2-3be6752c****</KeyId>
    <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
    <Plaintext>SGVsbG8gd29ybGQ=</Plaintext>
    <RequestId>475f1620-b9d3-4d35-b5c6-3fbdd941423d</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "KeyId": "5c438b18-05be-40ad-b6c2-3be6752c****",
  "KeyVersionId": "2ab1a983-7072-4bbc-a582-584b5bd8****",
  "Plaintext": "SGVsbG8gd29ybGQ=",
  "RequestId": "475f1620-b9d3-4d35-b5c6-3fbdd941423d"
}
```

错误码

访问错误中心查看更多错误码。

8.27. AsymmetricEncrypt

调用AsymmetricEncrypt接口使用非对称密钥进行加密。

仅支持**Usage**为**ENCRYPT / DECRYPT**的非对称密钥。支持的加密算法如下表：

KeySpec	Algorithm	说明	可加密的最大字节数
RSA_2048	RSAES_OAEP_SHA_256	RSAES-OAEP using SHA-256 and MGF1 with SHA-256	190
RSA_2048	RSAES_OAEP_SHA_1	RSAES-OAEP using SHA1 and MGF1 with SHA1	214
RSA_3072	RSAES_OAEP_SHA_256	RSAES-OAEP using SHA-256 and MGF1 with SHA-256	318
RSA_3072	RSAES_OAEP_SHA_1	RSAES-OAEP using SHA1 and MGF1 with SHA1	342
EC_SM2	SM2PKE	SM2椭圆曲线公钥加密算法	6047

本文将提供一个示例，使用密钥ID为 `5c438b18-05be-40ad-b6c2-3be6752c****` 、密钥版本ID为 `2ab1a983-7072-4bbc-a582-584b5bd8****` 的非对称密钥，通过加密算法 `RSAES_OAEP_SHA_1` 对明文 `SGVsbG8gd29ybGQ=` 进行加密。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	AsymmetricEncrypt	要执行的操作，取值：AsymmetricEncrypt。
Algorithm	String	是	RSAES_OAEP_SHA_1	加密算法。

名称	类型	是否必选	示例值	描述
KeyId	String	是	5c438b18-05be-40ad-b6c2-3be6752c****	主密钥（CMK）的全局唯一标识符。 说明 该参数也可以被指定为主密钥绑定的别名。更多信息，请参加见别名使用说明。
KeyVersionId	String	是	2ab1a983-7072-4bbc-a582-584b5bd8****	密钥版本ID。密钥版本的全局唯一标识符。 说明 您可以调用ListKeyVersions接口获取KeyVersionId（密钥版本ID）。
Plaintext	String	是	SGVsbG8gd29ybGQ=	要加密的明文，使用Base64编码。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
KeyId	String	5c438b18-05be-40ad-b6c2-3be6752c****	主密钥的全局唯一标识符。 说明 如果请求中的KeyId参数使用的是主密钥的别名，在响应中会返回别名对应的主密钥标识符。
KeyVersionId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	对明文数据进行加密的主密钥版本号。
CiphertextBlob	String	BQKP+1zK6+ZEMxTP5qaVzcsgXtWplYBKm0NXdSnB5FzliFxE1bSiu4dnEllca2JpeH7yz1/S6fed630H+hlH6DoM25fTLNcKj+mFB0Xnh9m2+HN59Mn4qyTfcUeadnfCXSWcGBouhXFwcdd2rJ3n337bzTf4jm659gZu3L0i6PLuxM9p7mqdwO0cKJPfGVfhnfMz+f4alMg79WB/NNyE2lyX7/qxvV49ObNrrJbKSFiz8Djocaf0IESNLMbfYI5bXjWkJlX92DQbKhibtQW8ZOJ//ZC6t0AWcUoKL6QDm/dg5koQalcleRinpB+QadFm894sLbVZ9+N4GVsv1Wbjwg==	加密后的密文，使用Base64编码。
RequestId	String	475f1620-b9d3-4d35-b5c6-3fbdd941423d	请求ID。

示例

请求示例

```
https://[Endpoint]/?Action=AsymmetricEncrypt
&KeyId=5c438b18-05be-40ad-b6c2-3be6752c****
&KeyVersionId=2ab1a983-7072-4bbc-a582-584b5bd8****
&Algorithm=RSAES_OAEP_SHA_1
&Plaintext=SGVsbG8gd29ybGQ=
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
    <KeyId>5c438b18-05be-40ad-b6c2-3be6752c****</KeyId>
    <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
    <CiphertextBlob>RKF5WeXJtusIrvuPOjpkA/55EKzi8Wmc/eJ2fQUKphvL750jtInSX1wijw/7jGxUaTHTW6tgIJl2ReN1aI1/wxqGxdzScwsMHxCBncnzQsZF+Fi4UFpI9pr4A1wc2u5Ngwyx9uA4K/kJ5bkS4NvmanxssAPZfSfbJSrAWlCP11tS0Cd54tQVGj4XK9tP9bJDKzKis1NClsOXZtNPX88kUqr3LkgFCsD07IwiePAfI2tn2fzeisje1Q7/d6VkF48c3ZE0DAmnLRujt3yRRGDaKUkI6SUDjuKD4yqBUX15/DKfJtya+JIPQGiO2IEPlhL7+NMT17U0tKtK5ZPNEwxfZw==</CiphertextBlob>
    <RequestId>475f1620-b9d3-4d35-b5c6-3fbdd941423d</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "KeyId": "5c438b18-05be-40ad-b6c2-3be6752c****",
  "KeyVersionId": "2ab1a983-7072-4bbc-a582-584b5bd8****",
  "CiphertextBlob": "RKF5WeXJtusIrvuPOjpkA/55EKzi8Wmc/eJ2fQUKphvL750jtInSX1wijw/7jGxUaTHTW6tgIJl2ReN1aI1/wxqGxdzScwsMHxCBncnzQsZF+Fi4UFpI9pr4A1wc2u5Ngwyx9uA4K/kJ5bkS4NvmanxssAPZfSfbJSrAWlCP11tS0Cd54tQVGj4XK9tP9bJDKzKis1NClsOXZtNPX88kUqr3LkgFCsD07IwiePAfI2tn2fzeisje1Q7/d6VkF48c3ZE0DAmnLRujt3yRRGDaKUkI6SUDjuKD4yqBUX15/DKfJtya+JIPQGiO2IEPlhL7+NMT17U0tKtK5ZPNEwxfZw==",
  "RequestId": "475f1620-b9d3-4d35-b5c6-3fbdd941423d"
}
```

错误码

访问错误中心查看更多错误码。

8.28. GetPublicKey

调用GetPublicKey接口获取非对称密钥的公钥。您可以在本地使用公钥进行加密、验签。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetPublicKey	要执行的操作，取值：GetPublicKey。
KeyId	String	是	5c438b18-05be-40ad-b6c2-3be6752c****	主密钥（CMK）的全局唯一标识符。该参数也可以被指定为CMK绑定的别名，详情见别名使用说明。
KeyVersionId	String	是	2ab1a983-7072-4bbc-a582-584b5bd8****	密钥版本的全局唯一标识符。

返回数据

名称	类型	示例值	描述
KeyId	String	5c438b18-05be-40ad-b6c2-3be6752c****	CMK的全局唯一标识符。 说明 如果请求中的KeyId参数使用的是CMK的别名，在响应中会返回别名对应的CMK标识符。
KeyVersionId	String	2ab1a983-7072-4bbc-a582-584b5bd8****	对明文数据进行加密的主密钥版本号。
PublicKey	String	-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs5Yu9AEgATN2/e3nUz1K\nEy6ng8MSPutcse2/VECG/NUF9C6D4IsJ64ShzY3dcn34WYzTOe916eMJFxyrNrSw\nHtc4UOR5AvaoRrfpgu2uq+i70/ZXrWL+pGb1hgZV8cWheIHMxwrR3liQlM5qN7EF\n9BdyWtyBfUGsp0Bn1VqlPc5G0x0a9xU2z9YtP994yDenNVIoIQ6Cov1lIEuwXAb2\n7boC41ePXwD0JWt41sP+rgCmpjBx00puIG+IlnoReEgI1ZGYmK98GgA/XzmNjZiD\nyvXJZAcM33Ue85+PkR5iHTtSEbi4QAoqpJabprUzz3Fin2j1dRrcacxGb7p31A9c\nJQIDAQAB\n-----END PUBLIC KEY-----\n	PEM格式的公钥。
RequestId	String	475f1620-b9d3-4d35-b5c6-3fbdd941423d	随机的访问ID。

示例

请求示例

```
https://[Endpoint]/?Action=GetPublicKey
&KeyId=5c438b18-05be-40ad-b6c2-3be6752c****
&KeyVersionId=2ab1a983-7072-4bbc-a582-584b5bd8****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
    <KeyId>5c438b18-05be-40ad-b6c2-3be6752c****</KeyId>
    <KeyVersionId>2ab1a983-7072-4bbc-a582-584b5bd8****</KeyVersionId>
    <PublicKey>-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs5Yu9AEgATN2/e3nUz1K\nEy6ng8MSPutcse2/VECG/NUF9C6D4IsJ64ShzY3dcn34WYzTOe916eMJFxyrNrSw\nHtc4UOR5AvaoRrfpgu2uq+i70/ZXrWL+pGb1hgZV8cWheIHMxwrR3IiQlM5qN7EF\n9BdyWtyBfUGsp0Bn1VqlPc5G0x0a9xU2z9YtP994yDenNVIoIQ6Cov1lIEuwXAb2\n7boC41ePXwD0JWt41sP+rgCmpjBx00puIG+IlnoReEgI1ZGYmK98GgA/XzmNjZiD\nyvXJZAcM33Ue85+PkR5iHTtSEbi4QAoqpJabprUzz3Fin2j1dRrcacxGb7p31A9c\nJQIDAQAB\n-----END PUBLIC KEY-----\n</PublicKey>
    <RequestId>475f1620-b9d3-4d35-b5c6-3fbdd941423d</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "KeyId": "5c438b18-05be-40ad-b6c2-3be6752c****",
  "KeyVersionId": "2ab1a983-7072-4bbc-a582-584b5bd8****",
  "PublicKey": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs5Yu9AEgATN2/e3nUz1K\nEy6ng8MSPutcse2/VECG/NUF9C6D4IsJ64ShzY3
dcn34WYzTOe916eMJFxyrNrSw\nHtc4UOR5AvaoRrfpgu2uq+i70/ZXrWL+pGb1hgZV8cWheIHMxwrR3IiQlM5qN7EF\n9BdyWtyBfUGsp0Bn1VqlPc5G0x0a9xU2z9YtP994yDenNVIoIQ6Cov1l
IEuwXAb2\n7boC41ePXwD0JWt41sP+rgCmpjBx00puIG+IlnoReEgI1ZGYmK98GgA/XzmNjZiD\nyvXJZAcM33Ue85+PkR5iHTtSEbi4QAoqpJabprUzz3Fin2j1dRrcacxGb7p31A9c\nJQIDAQA
B\n-----END PUBLIC KEY-----\n",
  "RequestId": "475f1620-b9d3-4d35-b5c6-3fbdd941423d"
}
```


错误码

访问错误中心查看更多错误码。

8.29. CreateAlias

调用CreateAlias接口为主密钥（CMK）创建一个别名。

使用说明：

- 每个别名只能表示一个CMK。
- 同一地域内的CMK别名必须唯一。

本文将提供一个示例，为密钥 `1234abcd-12ab-34cd-56ef-12345678****` 创建名为 `alias/example` 的别名。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateAlias	要执行的操作，取值：CreateAlias。
AliasName	String	是	alias/example	别名名称。 长度为1~255个字符，必须包含前缀 `alias/` 。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	CMK的全局唯一标识符。

返回数据

名称	类型	示例值	描述
RequestId	String	1d2baaf3-d357-46c2-832e-13560c2bd9cd	请求ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=CreateAlias
&AliasName=alias/example
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

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

`JSON` 格式

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

错误码

访问错误中心查看更多错误码。

8.30. UpdateAlias

调用UpdateAlias接口更新已存在的别名所代表的主密钥（CMK）ID。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateAlias	要执行的操作，取值：UpdateAlias。
AliasName	String	是	alias/example	要操作的别名。 长度为1~255个字符，必须包含前缀alias/。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	新的密钥ID。主密钥的全局唯一标识符。

返回数据

名称	类型	示例值	描述
RequestId	String	1d2baaf3-d357-46c2-832e-13560c2bd9cd	请求ID。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=UpdateAlias
&AliasName=alias/example
&KeyId=1234abcd-12ab-34cd-56ef-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

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

`JSON` 格式

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

错误码

HttpCode	错误码	错误信息	描述
400	Throttling	Request was denied due to request throttling.	您这个时段的流量已经超限。如果不能满足现有业务要求可以提工单进行申请。
404	Forbidden.KeyNotFound	The specified Key is not found.	指定的密钥不存在。

访问错误中心查看更多错误码。

8.31. DeleteAlias

调用DeleteAlias接口删除别名。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteAlias	要执行的操作，取值：DeleteAlias。
AliasName	String	是	alias/example	要操作的别名。 长度为1~255个字符，必须包含前缀alias/。

返回数据

名称	类型	示例值	描述
RequestId	String	4c8ae23f-3a42-6791-a4ba-1faa77831c28	请求ID。

示例

请求示例

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

正常返回示例

`XML` 格式

```
<KMS>
        <RequestId>4c8ae23f-3a42-6791-a4ba-1faa77831c28</RequestId>
</KMS>
```

`JSON` 格式

```
{
       "RequestId": "4c8ae23f-3a42-6791-a4ba-1faa77831c28"
}
```

错误码

访问错误中心查看更多错误码。

8.32. ListAliases

调用ListAliases接口查询当前用户在当前地域的所有别名。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListAliases	要执行的操作，取值：ListAliases。
PageNumber	Integer	否	1	当前页数。 取值范围：大于0的整数。 默认值：1。
PageSize	Integer	否	10	每页返回的结果个数。 取值范围：0~100。 默认值：10。

返回数据

名称	类型	示例值	描述
Aliases	Array of Alias		用户别名。
Alias			
AliasArn	String	acs:kms:cn-hangzhou:123456:alias/ExampleAlias1	别名的ARN。
AliasName	String	alias/ExampleAlias1	别名的唯一标识符。
KeyId	String	08c33a6f-4e0a-4a1b-a3fa-7ddfa1d****	别名对应的主密钥（CMK）。
PageNumber	Integer	1	当前页数。
PageSize	Integer	10	每页的返回结果个数。
RequestId	String	1b57992c-834b-4811-a889-f8bac1ba0353	请求ID。
TotalCount	Integer	1	返回的别名总数。

示例

请求示例

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

正常返回示例

`XML` 格式

```
<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>
```

`JSON` 格式

```
{
        "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"
}
```

错误码

访问错误中心查看更多错误码。

8.33. ListAliasesByKeyId

调用ListAliasesByKeyId查询与指定主密钥（CMK）对应的所有别名。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListAliasesByKeyId	系统规定参数。 取值：ListAliasesByKeyId。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	CMK的全局唯一标识符。
PageNumber	Integer	否	1	当前页数。 取值范围：大于0的整数。 默认值：1。
PageSize	Integer	否	10	每页返回的结果个数。 取值范围：0~101。 默认值：10。

返回数据

名称	类型	示例值	描述
TotalCount	Integer	1	返回的CMK总数。

名称	类型	示例值	描述
PageNumber	Integer	1	当前页数。
PageSize	Integer	10	每页的返回结果个数。
RequestId	String	1b57992c-834b-4811-a889-f8bac1ba0353	本次请求的ID。
Aliases	Array		别名。
KeyId	String	08c33a6f-4e0a-4a1b-a3fa-7ddfa1d4****	别名对应的CMK。
AliasName	String	alias/ExampleAlias1	别名的唯一标识符。
AliasArn	String	acs:kms:cn-hangzhou:123456:alias/ExampleAlias1	别名的ARN。

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>
```

正常返回示例

`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>
```

`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"
}
```

错误码

HttpCode	错误码	错误信息	描述
400	Throttling	Request was denied due to request throttling.	您这个时段的流量已经超限。如果不能满足现有业务要求可以提工单进行申请。

访问错误中心查看更多错误码。

9.凭据

9.1. CreateSecret

调用CreateSecret接口创建凭据，并存入凭据的初始版本。

您需要指定凭据名称、初始版本的凭据值和版本号。初始版本的状态被标记为ACSCurrent。

您可以指定一个对称密钥类型的用户主密钥（CMK）作为保护凭据的加密密钥。当不指定CMK时，凭据管家将为您自动创建一个主密钥，用于默认加密您的阿里云账号在本地域创建的凭据。凭据管家加密每个版本的凭据值，凭据名称、版本号、版本的状态标记等元数据不会被加密。

如果您指定主密钥，则需要同时具备相应主密钥的 `kms:GenerateDataKey` 权限，用于对凭据值进行加密。

本文将提供一个示例，创建一个名称为 `mydbconninfo` 、初始版本号 `VersionId` 为 `v1` 、凭据值 `SecretData` 为 `{"user":"root","passwd":"****"}` 的普通凭据。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateSecret	要执行的操作，取值：CreateSecret。
SecretData	String	是	{"user":"root","passwd":"****"}	新创建凭据的凭据值。凭据管家将其加密后，存入初始版本中。 • 当SecretType取值为Generic（普通凭据）时，您可以自定义凭据值。 • 当SecretType取值为Rds（托管RDS凭据）时，凭据值格式为：`{"Accounts":[{"AccountName":"","AccountPassword":""}]}` 。其中，`AccountName` 为RDS实例的账号名称，`AccountPassword` 为RDS实例的账号口令。 • 当SecretType取值为RAMCredentials（托管RAM凭据）时，凭据值格式为：`{"AccessKeys":[{"AccessKeyId":"","AccessKeySecret":"",}]}` 。其中，`AccessKeyId` 是访问密钥ID，`AccessKeySecret` 是访问密钥内容。您需要指定RAM用户的所有AccessKey。 • 当SecretType取值为ECS（托管ECS凭据）时，凭据值格式为： ◦ 当ExtendedConfig参数中SecretSubType取值为Password时：`{"UserName":"","Password": ""}` 。其中，`UserName` 为登录ECS实例的用户名，`Password` 为登录ECS实例的密码。 ◦ 当ExtendedConfig参数中SecretSubType取值为SSHKey时：`{"UserName":"","PublicKey": "", "PrivateKey": ""}` 。其中，`PublicKey` 为登录ECS实例的SSH格式公钥，`PrivateKey` 为登录ECS实例的私钥。
SecretName	String	是	mydbconninfo	凭据名称。 长度为1~64个字符，可包含英文字母、数字和特殊字符 `_/+=.@-` 。不同类型的凭据名称要求如下： • 当SecretType取值为Generic（普通凭据）或Rds（托管RDS凭据）时，不能以 `acs/` 开头。 • 当SecretType取值为RAMCredentials（托管RAM凭据）时，使用固定值 `$Auto` 。此时KMS自动生成凭据名称，以 `acs/ram/user/` 开头，包含RAM用户显示名称。 • 当SecretType取值为ECS（托管ECS凭据）时，必须以 `acs/ecs/` 开头。
VersionId	String	是	v1	初始版本的版本号。凭据对象内版本号唯一。
EncryptionKeyId	String	否	00aa68af-2c02-4f68-95fe-3435d330****	用于加密保护凭据值的KMS主密钥的标识符。 如果不指定，则凭据管家使用系统创建的密钥来加密保护凭据数据。 说明 KMS主密钥必须是对称密钥。
SecretDataType	String	否	text	凭据值类型。取值： • text • binary 说明 当SecretType取值为Rds、RAMCredentials或ECS时，SecretDataType取值只能为text。
Description	String	否	mydbinfo	凭据的描述信息。

名称	类型	是否必选	示例值	描述
Tags	String	否	[{\"TagKey\":\"key1\",\"TagValue\":\"val1\"},{\"TagKey\":\"key2\",\"TagValue\":\"val2\"}]	凭据的标签。
SecretType	String	否	Rds	凭据类型。取值： • Generic：普通凭据。 • Rds：托管RDS凭据。 • RAMCredentials：托管RAM凭据。 • ECS：托管ECS凭据。

ExtendedConfig	Json	否	{"SecretSubType":"SingleUser","DBInstanceId":"rm-bp1b3dd3a506e****","CustomData":{}}	凭据的拓展配置，用于指定特定凭据类型的属性。长度不超过1024个字符。
				• 当SecretType取值为Generic（普通凭据）时，忽略该参数。
				• 当SecretType取值为Rds（托管RDS凭据）时，需要指定ExtendedConfig的如下参数：
				  ○ SecretSubType（必填）：凭据子类型。取值：
				  ■ SingleUser：指定凭据管家以单账号模式托管RDS凭据。凭据轮转时，指定账号的口令会被重置为新的随机口令。
				  ■ DoubleUsers：指定凭据管家以双账号模式托管RDS凭据，ACSCurrent和ACSPrevious分别引用其中一个账号。凭据轮转时，ACSPrevious引用账号的口令会被重置为新的随机口令，随后凭据管家交换ACSCurrent和ACSPrevious对RDS账号的引用。
				  ○ DBInstanceId（必填）：指定RDS账号所在的RDS实例ID。
				  ○ CustomData（可选）：自定义数据。取值为JSON格式的键值对，最多不超过10个键值对，多个键值对用半角逗号（,）间隔。取值示例：`{"Key1": "v1", "fds":"fdsf"}`。默认值为空`{}`。
				• 当SecretType取值为RAMCredentials（托管RAM凭据）时，需要指定ExtendedConfig的如下参数：
				  ○ SecretSubType（必填）：凭据子类型。取值：RamUserAccessKey。
				  ○ UserName（必填）：RAM用户名称。
				  ○ CustomData（可选）：自定义数据。取值为JSON格式的键值对，最多不超过10个键值对，多个键值对用半角逗号（,）间隔。默认值为空`{}`。
				• 当SecretType取值为ECS（托管ECS凭据）时，需要指定ExtendedConfig的如下参数：
				  ○ SecretSubType（必填）：凭据子类型。取值：
				  ■ Password：ECS口令。
				  ■ SSHKey：ECS SSH公私钥。
				  ○ RegionId（必填）：ECS实例所在地域ID。
				  ○ InstanceId（必填）：ECS实例ID。
				  ○ CustomData（可选）：自定义数据。取值为JSON格式的键值对，最多不超过10个键值对，多个键值对用半角逗号（,）间隔。默认值为空`{}`。
				说明 当SecretType取值为Rds、RAMCredentials或ECS时，必须设置该参数。

EnableAutomaticRotation	Boolean	否	true	是否开启自动轮转，取值： • true：开启自动轮转。 • false（默认值）：不开启自动轮转。 **说明** 当SecretType取值为Rds、RAMCredentials或ECS时，该参数有效。

名称	类型	是否必选	示例值	描述
RotationInterval	String	否	30d	自动轮转的周期。取值范围：6小时~8,760小时（365天）。 格式为 `integer[unit]` ，其中 `integer` 表示时间长度， `unit` 表示时间单位。 unit取值：d（天）、h（小时）、m（分钟）、s（秒）。例如：7d或者604,800s均表示7天的周期。 说明 当EnableAutomaticRotation取值为true时，必须设置该参数。反之，将忽略该参数。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
Arn	String	acs:kms:cn-hangzhou:154035569884****:secret/mydbconninfo	阿里云资源名称。
AutomaticRotation	String	Enabled	是否开启自动轮转。取值： • Enabled：开启自动轮转。 • Disabled：不开启自动轮转。 • Invalid：轮转状态异常，凭据管家无法为您自动轮转。 说明 SecretType取值为Rds、RAMCredentials或ECS时，返回该参数。
ExtendedConfig	String	{\"SecretSubType\":\"SingleUser\",\"DBInstanceId\":\"rm-uf667446pc955****\",\"CustomData\":{}}	凭据的拓展配置。 说明 当SecretType取值为Rds、RAMCredentials或ECS时，返回该参数。
NextRotationDate	String	2020-07-06T18:22:03Z	下一次轮转的时间。 说明 当自动轮转开启时，返回该参数。
RequestId	String	3bf02f7a-015b-4f93-be0f-cc043fda2dd3	请求ID。
RotationInterval	String	604800s	凭据自动轮转的周期。 格式为 `integer[unit]` ，其中 `integer` 表示时间长度， `unit` 表示时间单位。 `unit` 取值：s（秒）。例如：7天的轮转周期为604800s。 说明 当自动轮转开启时，返回该参数。
SecretName	String	mydbconninfo	凭据名称。
SecretType	String	Rds	凭据类型。取值： • Generic：普通凭据。 • Rds：托管RDS凭据。 • RAMCredentials：托管RAM凭据。 • ECS：托管ECS凭据。
VersionId	String	v1	凭据版本号。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CreateSecret
&SecretData={"user":"root","passwd":"****"}
&SecretName=mydbconninfo
&VersionId=v1
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <Arn>acs:kms:cn-hangzhou:1540355698884****:secret/mydbconninfo</Arn>
   <SecretName>mydbconninfo</SecretName>
   <VersionId>v1</VersionId>
   <RequestId>3bf02f7a-015b-4f93-be0f-cc043fda2dd3</RequestId>
   <SecretType>Generic</SecretType>
 </KMS>
```

`JSON` 格式

```
{
 "Arn": "acs:kms:cn-hangzhou:1540355698884****:secret/mydbconninfo",
 "SecretName": "mydbconninfo",
 "VersionId": "v1",
 "RequestId": "3bf02f7a-015b-4f93-be0f-cc043fda2dd3",
 "SecretType": "Generic"
}
```

错误码

HttpCode	错误码	错误信息	描述
400	InvalidParameter	The specified parameter is invalid.	参数错误。
400	Rejected.LimitExceeded	The secret quota is exceeded.	凭据超出配额。
403	Forbidden.NoPermission	You are not authorized to perform the operation.	操作无权限。
404	Forbidden.ResourceNotFound	The resource is not found.	资源不存在。
409	Rejected.ResourceExist	The resource already exists.	资源已存在。
409	Rejected.ResourceInDeleteWindow	The secret is planned to be deleted.	此凭据在计划删除中。
500	InternalFailure	An internal error occurred.	内部错误。
429	Rejected.Throttling	The QPS upper limit is exceeded.	限流达到上限。

访问错误中心查看更多错误码。

9.2. ListSecrets

调用ListSecrets接口查询当前用户在当前地域创建的所有凭据。

此接口返回凭据对象的元数据信息，不返回被加密存储的凭据值。

本文将提供一个示例，返回当前用户在当前地域创建的凭据，其中当前页数`PageNumber`设置为`1`，每页中返回的个数`PageSize`设置为`2`，共返回2个凭据信息。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListSecrets	要执行的操作，取值：ListSecrets。
FetchTags	String	否	false	返回值中是否包含凭据的资源标签。取值： • true：包含。 • false（默认值）：不包含。
PageNumber	Integer	否	1	当前页数。 取值范围：大于0。 默认值：1。
PageSize	Integer	否	2	每页返回值的个数。 取值范围：1~100。 默认值：10。

名称	类型	是否必选	示例值	描述

Filters	String	否	[{"Key":"SecretName", "Values":["$Val1","$Val2"]}]	凭据过滤器。由Key-Values键值对组成，长度为0~1。使用一个标签键值过滤资源时，查询到的资源数量不能超过4000个。如果资源数量超过4000个，请使用ListResourceTags接口进行查询。
				• Key
				  ○ 描述：需要过滤的属性。
				  ○ 类型：String。
				  ○ 取值：
				  ■ SecretName：凭据名称。
				  ■ Description：凭据描述。
				  ■ TagKey：标签键。
				  ■ TagValue：标签值。
				• Values
				  ○ 描述：期望过滤后包含的值。
				  ○ 类型：String。
				  ○ 长度：0~10。
				  ○ 取值说明：
				  ■ Key取值为SecretName时：长度为1~192个字符，可包含英文字母、数字和特殊字符 `_/+=.@-` 。
				  ■ Key取值为Description时：长度为1~256个字符。
				  ■ Key取值为TagKey时：长度为1~256个字符，可包含英文字母、数字和特殊字符 `/_-.+=@:` 。
				  ■ Key取值为TagValue时：长度为1~256个字符，可包含英文字母、数字和特殊字符 `/_-.+=@:` 。
				Filters同一个Key中的多个Value之间的逻辑关系为OR。例如：输入 `[{"Key":"SecretName", "Values":["sec1","sec2"]}]` 时，语义为： `(SecretName=sec1 OR SecretName=sec2)` 。

返回数据

名称	类型	示例值	描述
PageNumber	Integer	1	当前页数。
PageSize	Integer	2	每页返回值的个数。
RequestId	String	6a6287a0-ff34-4780-a790-fdfca900557f	请求ID。
SecretList	Array of Secret		凭据列表。
Secret			
CreateTime	String	2020-07-17T07:59:05Z	创建时间。
PlannedDeleteTime	String	2020-08-17T07:59:05Z	计划删除时间。
SecretName	String	secret001	凭据名称。
SecretType	String	Generic	凭据类型。取值： • Generic：普通凭据。 • Rds：托管RDS凭据。
Tags	Array of Tag		凭据的资源标签。 如果FetchTags取值为false或者未指定，则不返回该参数。
Tag			
TagKey	String	key1	标签键。
TagValue	String	val1	标签值。
UpdateTime	String	2020-07-17T07:59:05Z	更新时间。
TotalCount	Integer	55	凭据列表中的凭据个数。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ListSecrets
&PageNumber=1
&PageSize=2
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <SecretList>
      <Secret>
         <SecretName>secret001</SecretName>
         <SecretType>Generic</SecretType>
         <CreateTime>2020-07-17T07:59:05Z</CreateTime>
         <UpdateTime>2020-07-17T07:59:05Z</UpdateTime>
      </Secret>
      <Secret>
         <SecretName>cache_client</SecretName>
         <SecretType>Generic</SecretType>
         <CreateTime>2020-07-23T11:56:29Z</CreateTime>
         <UpdateTime>2021-01-12T02:15:42Z</UpdateTime>
      </Secret>
   </SecretList>
   <RequestId>6a6287a0-ff34-4780-a790-fdfca900557f</RequestId>
   <PageNumber>1</PageNumber>
   <PageSize>2</PageSize>
   <TotalCount>55</TotalCount>
</KMS>
```

`JSON` 格式

```
{
 "SecretList": {
  "Secret": [
   {
    "SecretName": "secret001",
    "SecretType": "Generic",
    "CreateTime": "2020-07-17T07:59:05Z",
    "UpdateTime": "2020-07-17T07:59:05Z"
   },
   {
    "SecretName": "cache_client",
    "SecretType": "Generic",
    "CreateTime": "2020-07-23T11:56:29Z",
    "UpdateTime": "2021-01-12T02:15:42Z"
   }
  ]
 },
 "RequestId": "6a6287a0-ff34-4780-a790-fdfca900557f",
 "PageNumber": 1,
 "PageSize": 2,
 "TotalCount": 55
}
```

错误码

访问错误中心查看更多错误码。

9.3. DeleteSecret

调用DeleteSecret接口删除凭据对象。

默认为可恢复的删除，恢复窗口默认为30天。

您可以指定恢复窗口，也可以强制删除指定的凭据，且不允许恢复。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteSecret	要执行的操作，系统规定参数。 取值：**DeleteSecret**。
SecretName	String	是	secret001	凭据名称。

名称	类型	是否必选	示例值	描述
ForceDeleteWithoutRecovery	String	否	false	是否强制删除凭据，且不允许恢复。 取值范围： • true • false（默认值）
RecoveryWindowInDays	String	否	10	按照可恢复的方式删除凭据，且指定可恢复的窗口（天数）。默认值：30。

返回数据

名称	类型	示例值	描述
PlannedDeleteTime	String	2020-02-21T23:36:07.713703651+08:00	计划删除的时间戳。
RequestId	String	38bbed2a-15e0-45ad-98d4-816ad2ccf4ea	请求ID。
SecretName	String	secret001	删除的凭据名称。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DeleteSecret
&SecretName=secret001
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<SecretName>secret001</SecretName>
<RequestId>38bbed2a-15e0-45ad-98d4-816ad2ccf4ea</RequestId>
<PlannedDeleteTime>2020-02-21T23:36:07.713703651+08:00</PlannedDeleteTime>
```

`JSON` 格式

```
{
 "SecretName": "secret001",
 "RequestId": "38bbed2a-15e0-45ad-98d4-816ad2ccf4ea",
 "PlannedDeleteTime": "2020-02-21T23:36:07.713703651+08:00"
}
```

错误码

访问错误中心查看更多错误码。

9.4. DescribeSecret

调用DescribeSecret接口查询凭据的元数据信息。

此接口返回指定凭据的元数据信息，不返回被加密存储的凭据值。

本文将提供一个示例，查询一个名称为 `secret001` 凭据的元数据信息。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeSecret	要执行的操作，取值：DescribeSecret。
SecretName	String	是	secret001	凭据名称。
FetchTags	String	否	true	是否在返回参数中包含凭据的资源标签。取值： • true：包含资源标签。 • false（默认值）：不包含资源标签。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
Arn	String	acs:kms:cn-hangzhou:154035569884****:secret/secret001	凭据ARN。
AutomaticRotation	String	Enabled	是否开启自动轮转。取值： • Enabled：开启自动轮转。 • Disabled：不开启自动轮转。 • Invalid：轮转状态异常，凭据管家无法为您自动轮转。 说明 仅托管RDS凭据、托管RAM凭据或托管ECS凭据返回该参数。
CreateTime	String	2020-02-21T15:39:26Z	创建凭据的时间。
Description	String	userinfo	凭据的描述信息。
EncryptionKeyId	String	00aa68af-2c02-4f68-95fe-3435d330****	用于加密保护凭据值的KMS主密钥的标识符。
ExtendedConfig	String	{\"SecretSubType\":\"SingleUser\",\"DBInstanceId\":\"rm-uf667446pc955****\",\"CustomData\":{}}	凭据的拓展配置。 说明 仅托管RDS凭据、托管RAM凭据或托管ECS凭据返回该参数。
LastRotationDate	String	2020-07-05T08:22:03Z	最近一次轮转的时间。 说明 当凭据发生过轮转时返回该参数。
NextRotationDate	String	2020-07-06T18:22:03Z	下一次轮转的时间。 说明 当自动轮转开启时，返回该参数。
PlannedDeleteTime	String	2020-03-21T15:45:12Z	计划删除时间。
RequestId	String	93348dfb-3627-4417-8d90-487a76a909c9	请求ID。
RotationInterval	String	3153600s	凭据自动轮转的周期。 格式为 `integer[unit]` ，其中 `integer` 表示时间长度， `unit` 表示时间单位。 `unit` 取值：s（秒）。例如：7天的轮转周期为604800s。 说明 当自动轮转开启时，返回该参数。
SecretName	String	secret001	凭据名称。
SecretType	String	Rds	凭据类型。取值： • Generic：普通凭据。 • Rds：托管RDS凭据。 • RAMCredentials：托管RAM凭据。 • ECS：托管ECS凭据。
Tags	Array of Tag		凭据的资源标签。 如果FetchTags取值为false或者未指定，则不返回该参数。
Tag			
TagKey	String	key1	标签键。
TagValue	String	val1	标签值。
UpdateTime	String	2020-02-21T15:39:26Z	更新时间。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DescribeSecret
&SecretName=secret001
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <Arn>acs:kms:cn-hangzhou:154035569884****:secret/secret001</Arn>
   <SecretName>secret001</SecretName>
   <Description>userinfo</Description>
   <CreateTime>2021-01-08T10:50:05Z</CreateTime>
   <UpdateTime>2021-01-08T10:50:05Z</UpdateTime>
   <RequestId>93f84812-66d2-467a-9aec-61f6f53919dc</RequestId>
   <SecretType>Rds</SecretType>
   <AutomaticRotation>Disabled</AutomaticRotation>
   <ExtendedConfig>{"SecretSubType":"SingleUser", "DBInstanceId":"rm-uf667446pc955****",  "CustomData":{} }</ExtendedConfig>
</KMS>
```

`JSON` 格式

```
{
 "Arn": "acs:kms:cn-hangzhou:154035569884****:secret/secret001",
 "SecretName": "secret001",
 "Description": "userinfo",
 "CreateTime": "2021-01-08T10:50:05Z",
 "UpdateTime": "2021-01-08T10:50:05Z",
 "RequestId": "93f84812-66d2-467a-9aec-61f6f53919dc",
 "SecretType": "Rds",
 "AutomaticRotation": "Disabled",
 "ExtendedConfig": "{\"SecretSubType\":\"SingleUser\", \"DBInstanceId\":\"rm-uf667446pc955****\",  \"CustomData\":{} }"
}
```

错误码

访问错误中心查看更多错误码。

9.5. GetSecretValue

调用GetSecretValue接口获取凭据值。

如果不指定版本号或版本状态，则凭据管家默认返回被标记为ACSCurrent的版本凭据值。

如果凭据使用了用户指定的主密钥来保护凭据值，则需要调用者同时具备相应主密钥的 `kms:Decrypt` 权限。

本文将提供一个示例，获取一个名为 `secret001` 凭据的凭据值，返回结果显示凭据值 `SecretData` 为 `testdata1` 。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetSecretValue	要执行的操作，取值：GetSecretValue。
SecretName	String	是	secret001	凭据名称。
VersionStage	String	否	ACSCurrent	版本状态。如果指定该参数，则凭据管家返回被标记为指定状态的版本的凭据值。 默认值：ACSCurrent。 说明 托管RDS凭据、托管RAM凭据和托管ECS凭据只能获取ACSPrevious和ACSCurrent对应版本的凭据值。
VersionId	String	否	00000000000000000000000000000001	版本号。如果指定该参数，则凭据管家返回指定版本号的凭据值。 说明 托管RDS凭据、托管RAM凭据和托管ECS凭据不支持指定VersionId，设置该参数将被忽略。
FetchExtendedConfig	Boolean	否	true	是否获取凭据的拓展配置。取值： • true • false（默认值） 说明 普通凭据设置该参数将被忽略。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
AutomaticRotation	String	Enabled	是否开启自动轮转。取值： • Enabled：开启自动轮转。 • Disabled：不开启自动轮转。 • Invalid：轮转状态异常，凭据管家无法为您自动轮转。 说明 仅托管RDS凭据、托管RAM凭据或托管ECS凭据返回该参数。
CreateTime	String	2020-02-21T15:39:26Z	创建凭据的时间。
ExtendedConfig	String	{\"SecretSubType\":\"SingleUser\", \"DBInstanceId\":\"rm-uf667446pc955****\", \"CustomData\":{} }	凭据的拓展配置。 说明 当FetchExtendedConfig取值为true时，仅托管RDS凭据、托管RAM凭据或托管ECS凭据返回该参数。
LastRotationDate	String	2020-07-05T08:22:03Z	最近一次轮转的时间。 说明 当凭据发生过轮转时返回该参数。
NextRotationDate	String	2020-07-06T18:22:03Z	下一次轮转的时间。 说明 当自动轮转开启时，返回该参数。
RequestId	String	6a3e9c36-1150-4881-84d3-eb8672fcafad	请求ID。
RotationInterval	String	604800s	凭据自动轮转的周期。 格式为 `integer[unit]` ，其中 `integer` 表示时间长度， `unit` 表示时间单位。 `unit` 取值：s（秒）。例如：7天的轮转周期为604800s。 说明 当自动轮转开启时，返回该参数。

SecretData	String	testdata1	凭据值。凭据管家将存储的密文凭据值进行解密后返回该参数。
			• 通用凭据返回您指定的凭据值。
			• 托管RDS凭据返回的凭据值满足格式： `{"AccountName":"","AccountPassword":""}` 。
			• 托管RAM凭据返回的凭据值满足格式： `{"AccessKeyId":"Adfdsfd","AccessKeySecret":"fdsfdsf","GenerateTimestamp": "2016-03-25T10:42:40Z"}` 。
			• 托管ECS凭据返回的凭据值满足以下格式：
			  ◦ 当您托管ECS口令类型凭据时： `{"UserName":"root","Password":"H5asdasdsads****"}` 。
			  ◦ 当您托管ECS公私钥类型凭据时（私钥格式为PEM）： `{"UserName":"root","PublicKey":"ssh-rsa ****mKwnVix9YTFY9Rs= imported-openssh-key","PrivateKey": "d6bee1cb-2e14-4277-ba6b-73786b21****"}` 。

SecretDataType	String	binary	凭据值类型。取值： • text • binary
SecretName	String	secret001	凭据名称。
SecretType	String	Generic	凭据类型。取值： • Generic：普通凭据。 • Rds：托管RDS凭据。 • RAMCredentials：托管RAM凭据。 • ECS：托管ECS凭据。
VersionId	String	00000000000000000000000000000001	凭据版本的标识符。

名称	类型	示例值	描述
VersionStages	List	{ "VersionStage": ["ACSCurrent"] }	凭据版本的状态标记。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetSecretValue
&SecretName=secret001
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <SecretName>secret001</SecretName>
   <VersionId>00000000000000000000000000000001</VersionId>
   <SecretData>testdata1</SecretData>
   <SecretType>Generic</SecretType>
   <SecretDataType>binary</SecretDataType>
   <VersionStages>
      <VersionStage>ACSCurrent</VersionStage>
   </VersionStages>
   <CreateTime>2020-07-23T11:56:29Z</CreateTime>
   <RequestId>6a3e9c36-1150-4881-84d3-eb8672fcafad</RequestId>
</KMS>
```

`JSON` 格式

```
{
 "SecretName": "secret001",
 "VersionId": "00000000000000000000000000000001",
 "SecretData": "testdata1",
 "SecretType": "Generic",
 "SecretDataType": "binary",
 "VersionStages": {
  "VersionStage": [
   "ACSCurrent"
  ]
 },
 "CreateTime": "2020-07-23T11:56:29Z",
 "RequestId": "6a3e9c36-1150-4881-84d3-eb8672fcafad"
}
```

错误码

访问错误中心查看更多错误码。

9.6. PutSecretValue

调用PutSecretValue接口为凭据存入一个新版本的凭据值。

此接口用于存入新版本的凭据值，而不能用于修改已有版本的凭据值。

默认情况下，新存入的凭据值被标记为ACSCurrent，而ACSCurrent标记的前一个版本被标记为ACSPrevious。您可以通过指定VersionStage参数来覆盖该默认行为。

存入新版本时需指定版本号，凭据管家按照如下规则进行操作：

- 如果指定版本号在凭据内并不存在，则创建新版本并存入凭据值。
- 如果指定版本号在凭据内已经存在，且关联的凭据值和参数中指定的凭据值相等，则请求会被忽略，并且返回成功（请求是幂等的）。
- 如果指定版本号在凭据内已经存在，但是关联的凭据值和参数中指定的凭据值并不相等，则请求会被拒绝，且返回失败。

使用限制：该接口仅支持普通凭据。

本文将提供一个示例，为名为 `secret001` 的凭据存入一个新版本的凭据值，新凭据版本号 `VersionId` 为 `00000000000000000000000000000203` 、凭据值 `SecretData` 为 `importantdata` 。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	PutSecretValue	要执行的操作，取值：PutSecretValue。
SecretData	String	是	importantdata	凭据值。加密后存入指定的新版本中。
SecretName	String	是	secret001	凭据名称。

名称	类型	是否必选	示例值	描述
VersionId	String	是	00000000000000000000000000000000203	新凭据版本的版本号。凭据对象内版本号唯一。
SecretDataType	String	否	text	凭据值类型。取值： • text（默认值） • binary
VersionStages	String	否	{"VersionStage": ["ACSCurrent"] }	凭据版本在存入时需要被同时标记的版本状态。如果您不指定此参数，凭据管家默认为新版本标记ACSCurrent。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
RequestId	String	f94ec9d3-2d10-4922-9a5c-5dcd5ebcb5e8	请求ID。
SecretName	String	secret001	凭据名称。
VersionId	String	00000000000000000000000000000000203	被存入凭据版本的版本号。
VersionStages	List	{"VersionStage": ["ACSCurrent"]}	版本被标记的状态。

示例

请求示例

```
http(s)://[Endpoint]/?Action=PutSecretValue
&SecretData=importantdata
&SecretName=secret001
&VersionId=00000000000000000000000000000000203
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <SecretName>secret001</SecretName>
   <VersionId>00000000000000000000000000000000203</VersionId>
   <VersionStages>
      <VersionStage>ACSCurrent</VersionStage>
   </VersionStages>
   <RequestId>f94ec9d3-2d10-4922-9a5c-5dcd5ebcb5e8</RequestId>
</KMS>
```

`JSON` 格式

```
{
 "SecretName": "secret001",
 "VersionId": "00000000000000000000000000000000203",
 "VersionStages": {
  "VersionStage": [
   "ACSCurrent"
  ]
 },
 "RequestId": "f94ec9d3-2d10-4922-9a5c-5dcd5ebcb5e8"
}
```

错误码

访问错误中心查看更多错误码。

9.7. UpdateSecret

调用UpdateSecret接口更新凭据的元数据。

本文将提供一个示例，更新名为 `secret001` 凭据的元数据，将其描述信息 `Description` 更新为 `datainfo` 。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateSecret	要执行的操作，取值：UpdateSecret。
SecretName	String	是	secret001	凭据名称。
Description	String	否	datainfo	凭据的描述信息。
ExtendedConfig.CustomData	Json	否	{"DBName":"app1","Port":"3306"}	拓展配置中的自定义数据。 说明 • 如果指定该参数，将会更新凭据已有的拓展配置。 • 普通凭据不支持设置该参数。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
RequestId	String	5b75d8b1-5b6a-4ec0-8e0c-c08befdfad47	请求ID。
SecretName	String	secret001	凭据名称。

示例

请求示例

```
http(s)://[Endpoint]/?Action=UpdateSecret
&Description=datainfo
&SecretName=secret001
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <SecretName>secret001</SecretName>
   <RequestId>5b75d8b1-5b6a-4ec0-8e0c-c08befdfad47</RequestId>
</KMS>
```

`JSON` 格式

```
{
    "SecretName":"secret001",
    "RequestId": "5b75d8b1-5b6a-4ec0-8e0c-c08befdfad47"
}
```

错误码

访问错误中心查看更多错误码。

9.8. UpdateSecretVersionStage

调用UpdateSecretVersionStage接口更新凭据的版本状态。

该接口用于更新凭据的版本状态，支持以下两种操作方式：

- 将指定的版本状态用于标记一个新的凭据版本。
- 将指定的版本状态从被标记的凭据版本上移除。

使用限制：该接口仅支持普通凭据。

本文将提供一个示例，更新名为 `secret001` 凭据的版本状态，将 `ACSCurrent` 版本状态用于标记 `002` 版本。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateSecretVersionStage	要执行的操作，取值：UpdateSecretVersionStage。

名称	类型	是否必选	示例值	描述
SecretName	String	是	secret001	凭据名称。
VersionStage	String	是	ACSCurrent	指定版本状态。取值： • ACSCurrent • ACSPrevious • 自定义状态
RemoveFromVersion	String	否	001	将指定的版本状态从此参数指定的版本上移除。 说明 RemoveFromVersion和MoveToVersion至少指定其中一个参数。
MoveToVersion	String	否	002	将指定的版本状态用于标记此参数指定的版本。 说明 • RemoveFromVersion和MoveToVersion至少指定其中一个参数。 • 当VersionStage取值为ACSCurrent或ACSPrevious时，必须指定该参数。

返回数据

名称	类型	示例值	描述
RequestId	String	8cad259f-4d77-40ec-bbd7-b9c47a423bb9	请求ID。
SecretName	String	secret001	凭据名称。

示例

请求示例

```
http(s)://[Endpoint]/?Action=UpdateSecretVersionStage
&SecretName=secret001
&VersionStage=ACSCurrent
&MoveToVersion=002
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <SecretName>secret001</SecretName>
   <RequestId>8cad259f-4d77-40ec-bbd7-b9c47a423bb9</RequestId>
</KMS>
```

`JSON` 格式

```
{
 "SecretName": "secret001",
 "RequestId": "8cad259f-4d77-40ec-bbd7-b9c47a423bb9"
}
```

错误码

访问错误中心查看更多错误码。

9.9. RestoreSecret

调用RestoreSecret接口恢复被删除的凭据。

此接口只能在可恢复的窗口期内，用于恢复被删除的凭据。如果删除凭据时指定了**ForceDeleteWithoutRecovery**参数为**true**，则不能调用此接口来恢复。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述

名称	类型	是否必选	示例值	描述
Action	String	是	RestoreSecret	要执行的操作，系统规定参数。 取值：**RestoreSecret**。
SecretName	String	是	secret001	凭据名称。

返回数据

名称	类型	示例值	描述
RequestId	String	e4885adf-548f-4ca5-8075-f540bbd3a55f	请求ID。
SecretName	String	secret001	凭据名称。

示例

请求示例

```
http(s)://[Endpoint]/?Action=RestoreSecret
&SecretName=secret001
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<RequestId>e4885adf-548f-4ca5-8075-f540bbd3a55f</RequestId>
<SecretName>secret001</SecretName>
```

`JSON` 格式

```
{
 "RequestId": "e4885adf-548f-4ca5-8075-f540bbd3a55f",
 "SecretName": "secret001"
}
```

错误码

访问错误中心查看更多错误码。

9.10. ListSecretVersionIds

调用ListSecretVersionIds接口查询凭据的所有版本信息。

版本信息中不包含凭据值。默认不返回可回收的（Deprecated）凭据版本。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListSecretVersionIds	要执行的操作，系统规定参数。 取值：**ListSecretVersionIds**。
SecretName	String	是	secret001	凭据名称。
IncludeDeprecated	String	否	false	返回值中是否包含可回收的凭据版本。 默认值：**false**。
PageNumber	Integer	否	1	当前页。
PageSize	Integer	否	10	当前页的大小。

返回数据

名称	类型	示例值	描述
PageNumber	Integer	1	当前页。

名称	类型	示例值	描述
PageSize	Integer	10	当前页的大小。
RequestId	String	5b75d8b1-5b6a-4ec0-8e0c-c08befdfad47	请求ID。
SecretName	String	secret001	凭据名称。
TotalCount	Integer	4	当前页返回的列表项个数。
VersionIds	Array		凭据的版本信息列表。
CreateTime	String	2020-02-21T15:39:26Z	版本的创建时间。
VersionId	String	00000000000000000000000000000000000203	版本号。
VersionStages	List	{ "VersionStage": ["ACSCurrent", "UStage1", "Ustage2"] }	版本的状态标记。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ListSecretVersionIds
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<SecretName>secret001</SecretName>
<VersionIds>
    <VersionId>
        <VersionId>00000000000000000000000000000203</VersionId>
        <VersionStages>
            <VersionStage>ACSCurrent</VersionStage>
            <VersionStage>UStage1</VersionStage>
            <VersionStage>Ustage2</VersionStage>
        </VersionStages>
    </VersionId>
    <VersionId>
        <VersionId>00000000000000000000000000000001</VersionId>
        <VersionStages>
            <VersionStage>ACSPrevious</VersionStage>
        </VersionStages>
    </VersionId>
</VersionIds>
<RequestId>5b75d8b1-5b6a-4ec0-8e0c-c08befdfad47</RequestId>
<PageNumber>1</PageNumber>
<PageSize>10</PageSize>
<TotalCount>4</TotalCount>
```

`JSON` 格式

```
{
    "SecretName": "secret001",
    "VersionIds": {
        "VersionId": [
            {
                "VersionId": "00000000000000000000000000000203",
                "VersionStages": {
                    "VersionStage": [
                        "ACSCurrent",
                        "UStage1",
                        "Ustage2"
                    ]
                }
            },
            {
                "VersionId": "00000000000000000000000000000001",
                "VersionStages": {
                    "VersionStage": [
                        "ACSPrevious"
                    ]
                }
            }
        ]
    },
    "RequestId": "5b75d8b1-5b6a-4ec0-8e0c-c08befdfad47",
    "PageNumber": 1,
    "PageSize": 10,
    "TotalCount": 4
}
```


错误码

访问错误中心查看更多错误码。

9.11. GetRandomPassword

调用GetRandomPassword接口获得一个随机口令字符串。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetRandomPassword	要执行的操作，取值：GetRandomPassword。
PasswordLength	String	否	32	生成口令的字节数。 取值：8~128。 默认值：32。
ExcludeCharacters	String	否	ABCabc	生成口令时排除的字符。 有效值： `0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!\"#$%&'()*+,-./:;<=>?@[\]^_{|}~` 。 默认值：空。
ExcludeLowercase	String	否	false	生成口令时是否排除小写字母。 取值： • true • false（默认值）
ExcludeUppercase	String	否	false	生成口令时是否排除大写字母。 取值： • true • false（默认值）
ExcludeNumbers	String	否	false	生成口令时是否排除数字。 取值： • true • false（默认值）

名称	类型	是否必选	示例值	描述
ExcludePunctuation	String	否	false	生成口令时是否排除特殊字符。 取值： • true • false（默认值）
RequireEachIncludedType	String	否	true	生成口令时是否上述每种类型都包含。 取值： • true • false（默认值）

返回数据

名称	类型	示例值	描述
RandomPassword	String	IxGn>NMmNB(y?iZ<Yc,_H/{2GC'U****	随机口令。
RequestId	String	6b0cbe25-5e33-467e-972e-7a83c6c97604	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetRandomPassword
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<RequestId>6b0cbe25-5e33-467e-972e-7a83c6c97604</RequestId>
<RandomPassword>IxGn&gt;NMmNB(y?iZ&lt;Yc,_H/{2GC'U****</RandomPassword>
```

`JSON` 格式

```
{"RequestId":"6b0cbe25-5e33-467e-972e-7a83c6c97604","RandomPassword":"IxGn>NMmNB(y?iZ<Yc,_H/{2GC'U****"}
```

错误码

访问错误中心查看更多错误码。

9.12. RotateSecret

调用RotateSecret接口手动轮转凭据。

使用限制：

• 同一阿里云账号每小时最多轮转50次。

• RotateSecret接口不支持轮转普通凭据。

本文将提供一个示例，将名称为 `RdsSecret/Mysql5.4/MyCred` 的凭据进行手动轮转，轮转后新的凭据版本为 `000000123` 。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	RotateSecret	要执行的操作，取值：RotateSecret。
SecretName	String	是	RdsSecret/Mysql5.4/MyCred	凭据名称。
VersionId	String	是	000000123	轮转后的凭据新版本的版本号。 **说明** 版本号用于保证请求的幂等性。凭据管家使用版本号来防止您的应用在请求失败后进行重试时，意外创建重复的版本。如果相同的版本号已经存在，轮转的请求会被忽略，服务端会返回成功。

返回数据

名称	类型	示例值	描述
Arn	String	acs:kms:cn-hangzhou:154035569884****:secret/RdsSecret/Mysql5.4/MyCred	凭据ARN。
SecretName	String	RdsSecret/Mysql5.4/MyCred	凭据名称。
VersionId	String	000000123	轮转后的凭据新版本的版本号。
RequestId	String	10257c86-269d-43aa-aaf3-90ed4144bb7c	请求ID。

关于公共请求参数的详情，请参见公共参数。

示例

请求示例

```
http(s)://[Endpoint]/?Action=RotateSecret
&SecretName=RdsSecret/Mysql5.4/MyCred
&VersionId=000000123
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <Arn>acs:kms:cn-hangzhou:154035569884****:secret/RdsSecret/Mysql5.4/MyCred</Arn>
   <SecretName>RdsSecret/Mysql5.4/MyCred</SecretName>
   <VersionId>000000123</VersionId>
   <RequestId>10257c86-269d-43aa-aaf3-90ed4144bb7c</RequestId>
</KMS>
```

`JSON` 格式

```
{
 "Arn": "acs:kms:cn-hangzhou:154035569884****:secret/RdsSecret/Mysql5.4/MyCred",
 "SecretName": "RdsSecret/Mysql5.4/MyCred",
 "VersionId": "000000123",
 "RequestId": "10257c86-269d-43aa-aaf3-90ed4144bb7c"
}
```

错误码

访问错误中心查看更多错误码。

9.13. UpdateSecretRotationPolicy

调用UpdateSecretRotationPolicy接口更新凭据轮转策略。

凭据开启自动轮转后，首次自动轮转的时间为上次轮转时间加上轮转周期。

使用限制：UpdateSecretRotationPolicy接口不支持更新普通凭据的轮转策略。

本文将提供一个示例，更新名称为 `RdsSecret/Mysql5.4/MyCred` 的凭据轮转策略。具体如下：

- 将 `EnableAutomaticRotation` 设置为 `true` ，表示开启自动轮转。
- 将 `RotationInterval` 设置为 `30d` ，表示设置轮转周期为30天。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateSecretRotationPolicy	要执行的操作，取值：UpdateSecretRotationPolicy。
EnableAutomaticRotation	Boolean	是	true	是否开启自动轮转，取值： • true：开启自动轮转。 • false（默认值）：不开启自动轮转。
SecretName	String	是	RdsSecret/Mysql5.4/MyCred	凭据名称。

名称	类型	是否必选	示例值	描述
RotationInterval	String	否	30d	自动轮转的周期。取值范围：6小时~8,760小时（365天）。 格式为 `integer[unit]` ，其中 `integer` 表示时间长度， `unit` 表示时间单位。 unit取值：d（天）、h（小时）、m（分钟）、s（秒）。例如：7d或者604,800s均表示7天的周期。 **说明** 当EnableAutomaticRotation取值为true时，必须设置该参数。反之，将忽略该参数。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
SecretName	String	RdsSecret/Mysql5.4/MyCred	凭据名称。
RequestId	String	2c124f6f-4210-499f-b88a-69f54004d2d8	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=UpdateSecretRotationPolicy
&EnableAutomaticRotation=true
&SecretName=RdsSecret/Mysql5.4/MyCred
&RotationInterval=30d
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <SecretName>RdsSecret/Mysql5.4/MyCred</SecretName>
   <RequestId>2c124f6f-4210-499f-b88a-69f54004d2d8</RequestId>
</KMS>
```

`JSON` 格式

```
{
    "SecretName": "RdsSecret/Mysql5.4/MyCred",
    "RequestId": "2c124f6f-4210-499f-b88a-69f54004d2d8"
}
```

错误码

访问错误中心查看更多错误码。

10.证书

10.1. CreateCertificate

调用CreateCertificate接口创建证书。

创建证书时，请指定非对称密钥的类型，将私钥托管在证书管家，并获取证书请求CSR（Certificate Signing Request）。然后将PEM格式的证书请求文件提交给CA机构，获取正式的证书和证书链，并调用UploadCertificate接口导入证书管家。

本文将提供一个示例，创建一个证书，并获取证书请求CSR。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateCertificate	要执行的操作，取值：CreateCertificate。
KeySpec	String	是	RSA_2048	密钥类型，取值： • RSA_2048 • EC_P256 • EC_SM2
Subject	String	是	CN=userName,OU=kms,O=aliyun,C=CN	证书主体（拥有者）。 按照RFC 2253定义，采用DN（Distinguished Names）标识。DN由一系列RDN（Relative Distinguished Names）组成。 RDN是一组键值对，多个RDN之间用英文逗号（,）隔开，格式为：`attribute1=value1,attribute2=value2`。 证书主体字段含义如下： • CN（必选）：名称。证书使用主体名称。 • C（必选）：国家/地区。使用ISO 3166-1的二位国家代码。例如：CN代表中国。 • O（必选）：公司名称。企业、单位、组织或机构的法定名称。 • OU（必选）：部门名称。 • ST（可选）：省/市。省、直辖市、自治区或特别行政区名称。 • L（可选）：城市名称。
SubjectAlternativeNames	Json	否	["test1.example.com","test2.example.com"]	证书主体别名。 支持域名列表，最多支持10个域名。
ExportablePrivateKey	Boolean	否	true	证书私钥是否需要导出使用。取值： • true（默认值）：证书私钥需要导出使用。 • false：证书私钥不需要导出使用。建议选择否，以便使用更高安全级别的密钥保护。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
Arn	String	acs:kms:cn-hangzhou:154035569884****:certificate/98e85c94-52d0-40c9-b3b2-afda52f4****	证书ARN。
CertificateId	String	9a28de48-8d8b-484d-a766-dec4****	证书ID。证书管家中证书的全局唯一标识符。
Csr	String	-----BEGIN CERTIFICATE REQUEST-----\nMIIDADCCAegCAQAwgboxCzAJBgNVBAYTAkNOMREwDwYDVQQIEwhaaGVqaWFuZzER\n****\nmkj4rg==\n-----END CERTIFICATE REQUEST-----\n	PEM格式的证书请求。
RequestId	String	15a735a1-8fe6-45cc-a64c-3c4ff839334e	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CreateCertificate
&KeySpec=RSA_2048
&Subject=CN=userName,OU=kms,O=aliyun,C=CN
&<公共请求参数>
```

正常返回示例

`JSON` 格式

```
{
 "CertificateId": "98e85c94-52d0-40c9-b3b2-afda52f4****",
 "Arn": "acs:kms:cn-hangzhou:154035569884****:certificate/98e85c94-52d0-40c9-b3b2-afda52f4****",
 "Csr": "-----BEGIN CERTIFICATE REQUEST-----\nMIIDADCCAegCAQAwgboxCzAJBgNVBAYTAkNOMREwDwYDVQQIEwhaaGVqaWFuZzER\n****\nmkj4rg==\n-----END CERTIFICATE
REQUEST-----\n",
 "RequestId": "15a735a1-8fe6-45cc-a64c-3c4ff839334e"
}
```

错误码

HttpCode	错误码	错误信息	描述
404	InvalidAccessKeyId.NotFound	The specified AccessKey ID does not exist.	AccessKey ID找不到。请检查调用时是否使用了正确的AccessKey。

访问错误中心查看更多错误码。

10.2. UploadCertificate

调用UploadCertificate接口将CA机构颁发的证书和证书链导入证书管家。

本文将提供一个示例，将CA机构颁发的证书导入到KMS证书管家中ID为 `12345678-1234-1234-1234-12345678****` 的证书中。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UploadCertificate	要执行的操作，取值：UploadCertificate。
Certificate	String	是	-----BEGIN CERTIFICATE----- (X.509 Certificate PEM Content) -----END CERTIFICATE-----	CA机构颁发的PEM格式的证书。
CertificateId	String	是	12345678-1234-1234-1234-12345678****	证书ID。证书管家中证书的全局唯一标识符。
CertificateChain	String	否	-----BEGIN CERTIFICATE----- (Sub CA Certificate PEM Content) -----END CERTIFICATE----- -----BEGIN CERTIFICATE----- (Sub CA Certificate PEM Content) -----END CERTIFICATE----- -----BEGIN CERTIFICATE----- (Root CA Certificate PEM Content) -----END CERTIFICATE-----	CA机构颁发的PEM格式的证书链。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
RequestId	String	15a735a1-8fe6-45cc-a64c-3c4ff839334e	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=UploadCertificate
&Certificate=-----BEGIN CERTIFICATE-----  (X.509 Certificate PEM Content)  -----END CERTIFICATE-----
&CertificateId=12345678-1234-1234-1234-12345678****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <RequestId>15a735a1-8fe6-45cc-a64c-3c4ff839334e</RequestId>
</KMS>
```

`JSON` 格式

```
{
    "RequestId": "15a735a1-8fe6-45cc-a64c-3c4ff839334e"
}
```

错误码

HttpCode	错误码	错误信息	描述
404	InvalidAccessKeyId.NotFound	The specified AccessKey ID does not exist.	AccessKey ID找不到。请检查调用时是否使用了正确的AccessKey。

访问错误中心查看更多错误码。

10.3. GetCertificate

调用GetCertificate接口查询证书管家托管的证书。

本文将提供一个示例，查询ID为 `9a28de48-8d8b-484d-a766-dec4****` 的证书信息，包括证书、证书链、证书ID和证书请求。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetCertificate	要执行的操作，取值：GetCertificate。
CertificateId	String	是	9a28de48-8d8b-484d-a766-dec4****	证书ID。证书管家中证书的全局唯一标识符。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
Certificate	String	-----BEGIN CERTIFICATE----- (X.509 Certificate PEM Content) -----END CERTIFICATE-----	PEM格式的证书。
CertificateChain	String	-----BEGIN CERTIFICATE----- (Sub CA Certificate PEM Content) -----END CERTIFICATE----- -----BEGIN CERTIFICATE----- (Sub CA Certificate PEM Content) -----END CERTIFICATE----- -----BEGIN CERTIFICATE----- (Root CA Certificate PEM Content) -----END CERTIFICATE-----	PEM格式的证书链。
CertificateId	String	9a28de48-8d8b-484d-a766-dec4****	证书ID。
Csr	String	-----BEGIN CERTIFICATE REQUEST----- MIICxjCCAa4CAQAwPzELMAkGA1UEBhMCQ04xDzANBgNVBAoTBmFsaXl1bjEMMAoGA1UECxMDa21zMREwDwY****-----END CERTIFICATE REQUEST-----	PEM格式的证书请求。
RequestId	String	b3e104b4-0319-4a20-ab7f-9fef6****	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetCertificate
&CertificateId=9a28de48-8d8b-484d-a766-dec4****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <Csr>-----BEGIN CERTIFICATE REQUEST-----MIICxjCCAa4CAQAwPzELMAkGA1UEBhMCQ04xDzANBgNVBAoTBmFsaXl1bjEMMAoGA1UECxMDa21zMREwDwY****-----END CERTIFICATE REQUEST-----</Csr>
   <RequestId>b3e104b4-0319-4a20-ab7f-9fef6****</RequestId>
   <CertificateId>9a28de48-8d8b-484d-a766-dec4****</CertificateId>
   <CertificateChain>-----BEGIN CERTIFICATE-----  (Sub CA Certificate PEM Content)  -----END CERTIFICATE-----  -----BEGIN CERTIFICATE-----  (Sub CA Certificate PEM Content)  -----END CERTIFICATE-----  -----BEGIN CERTIFICATE-----  (Root CA Certificate PEM Content)  -----END CERTIFICATE-----</CertificateChain>
   <Certificate>-----BEGIN CERTIFICATE-----  (X.509 Certificate PEM Content)  -----END CERTIFICATE-----</Certificate>
</KMS>
```

`JSON` 格式

```
{
 "Csr": "-----BEGIN CERTIFICATE REQUEST-----MIICxjCCAa4CAQAwPzELMAkGA1UEBhMCQ04xDzANBgNVBAoTBmFsaXl1bjEMMAoGA1UECxMDa21zMREwDwY****-----END CERTIFICATE REQUEST-----",
 "RequestId": "b3e104b4-0319-4a20-ab7f-9fef6****",
 "CertificateId": "9a28de48-8d8b-484d-a766-dec4****",
 "CertificateChain": "-----BEGIN CERTIFICATE-----  (Sub CA Certificate PEM Content)  -----END CERTIFICATE-----  -----BEGIN CERTIFICATE-----  (Sub CA Certificate PEM Content)  -----END CERTIFICATE-----  -----BEGIN CERTIFICATE-----  (Root CA Certificate PEM Content)  -----END CERTIFICATE-----",
 "Certificate": "-----BEGIN CERTIFICATE-----  (X.509 Certificate PEM Content)  -----END CERTIFICATE-----"
}
```

错误码

HttpCode	错误码	错误信息	描述
404	InvalidAccessKeyId.NotFound	The specified AccessKey ID does not exist.	AccessKey ID找不到。请检查调用时是否使用了正确的AccessKey。

访问错误中心查看更多错误码。

10.4. DescribeCertificate

调用DescribeCertificate接口查询证书信息。

本文将提供一个示例，查询一个ID为 `9a28de48-8d8b-484d-a766-dec4****` 的证书信息，包括证书ID、创建时间、签发者信息、有效期、序列号、签名算法等。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeCertificate	要执行的操作，取值：DescribeCertificate。
CertificateId	String	是	9a28de48-8d8b-484d-a766-dec4****	证书ID。证书管家中证书的全局唯一标识符。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
Arn	String	acs:kms:cn-hangzhou:159498693826****:certificate/9a28de48-8d8b-484d-a766-dec4****"	证书ARN。
CertificateId	String	9a28de48-8d8b-484d-a766-dec4****	证书ID。证书管家中证书的全局唯一标识符。
CreatedAt	String	2020-10-13T03:05:03Z	证书的创建时间。
ExportablePrivateKey	Boolean	true	证书密钥是否支持导出。取值： • true（默认值）：支持导出。 • false：不支持导出。
Issuer	String	CN=testCA,OU=kms,O=aliyun,C=CN	证书的签发者信息，使用限定名DN（Distinguished Names）形式标识。
KeySpec	String	RSA_2048	密钥类型。
NotAfter	String	2022-10-13T03:09:00Z	证书有效期的截止时间。

名称	类型	示例值	描述
NotBefore	String	2020-10-13T03:09:00Z	证书有效期的开始时间。
RequestId	String	edb671a3-c5a1-4ebe-a1de-d748b640bdf2	请求ID。
Serial	String	12345678	证书序列号。
SignatureAlgorithm	String	ECDSA-SHA256	证书签名算法，取值： • RSA2048-SHA256 • ECDSA-SHA256 • SM2-SM3
Status	String	ACTIVE	证书的状态信息，取值： • PENDING：等待导入。 • ACTIVE：已启用。 • INACTIVE：已禁用。 • REVOKED：已吊销。
Subject	String	CN=userName,OU=aliyun,O=aliyun,C=CN	证书主体（拥有者），采用DN标识。
SubjectAlternativeNames	List	["test1.example.com","test2.example.com"]	证书主体别名。 支持域名列表，最多支持10个域名。
SubjectKeyIdentifier	String	79 36 26 DE 9F F5 15 E3 56 DC ****	主体公钥识别符。
SubjectPublicKey	String	-----BEGIN PUBLIC KEY----- MIIBIjA -----END PUBLIC KEY-----	证书中的公钥。
Tags	Map	[{\"TagKey\":\"S1key1\",\"TagValue\":\"S1val1\"},{\"TagKey\":\"S1key2\",\"TagValue\":\"S2val2\"}]	证书对应的标签。
UpdatedAt	String	2020-12-23T06:10:13Z	证书的更新时间。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DescribeCertificate
&CertificateId=9a28de48-8d8b-484d-a766-dec4****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
    <Status>ACTIVE</Status>
    <RequestId>edb671a3-c5a1-4ebe-a1de-d748b640bdf2</RequestId>
    <Issuer>CN=testCA,OU=kms,O=aliyun,C=CN</Issuer>
    <CertificateId>9a28de48-8d8b-484d-a766-dec4****</CertificateId>
    <CreatedAt>2020-10-13T03:05:03Z</CreatedAt>
    <KeySpec>RSA_2048</KeySpec>
    <SubjectAlternativeNames>[\"test1.example.com\",\"test2.example.com\"]</SubjectAlternativeNames>
    <SignatureAlgorithm>ECDSA-SHA256</SignatureAlgorithm>
    <SubjectKeyIdentifier>79 36 26 DE 9F F5 15 E3 56 DC ********</SubjectKeyIdentifier>
    <NotAfter>2022-10-13T03:09:00Z</NotAfter>
       <ExportablePrivateKey>true</ExportablePrivateKey>
    <UpdatedAt>2020-10-13T03:15:00Z</UpdatedAt>
    <Subject>CN=userName,OU=aliyun,O=aliyun,C=CN</Subject>
    <Serial>12345678</Serial>
    <SubjectPublicKey>-----BEGIN PUBLIC KEY----- MIIBIjA -----END PUBLIC KEY-----</SubjectPublicKey>
    <NotBefore>2020-10-13T03:09:00Z</NotBefore>
    <Arn>acs:kms:cn-hangzhou:159498693826****:certificate/9a28de48-8d8b-484d-a766-dec4****</Arn>
    <Tags>[{\"TagKey\":\"S1key1\",\"TagValue\":\"S1val1\"},{\"TagKey\":\"S1key2\",\"TagValue\":\"S2val2\"}]</Tags>
</KMS>
```

`JSON` 格式

```
{
    "Status": "ACTIVE",
    "RequestId": "edb671a3-c5a1-4ebe-a1de-d748b640bdf2",
    "Issuer": "CN=testCA,OU=kms,O=aliyun,C=CN",
    "CertificateId": "9a28de48-8d8b-484d-a766-dec4****",
    "CreatedAt": "2020-10-13T03:05:03Z",
    "KeySpec": "RSA_2048",
    "SubjectAlternativeNames": "[\"test1.example.com\",\"test2.example.com\"]",
    "SignatureAlgorithm": "ECDSA-SHA256",
    "SubjectKeyIdentifier": "79 36 26 DE 9F F5 15 E3 56 DC ********",
    "NotAfter": "2022-10-13T03:09:00Z",
    "ExportablePrivateKey": "true",
    "UpdatedAt": "2020-10-13T03:15:00Z",
    "Subject": "CN=userName,OU=aliyun,O=aliyun,C=CN",
    "Serial": "12345678",
    "SubjectPublicKey": "-----BEGIN PUBLIC KEY----- MIIBIjA -----END PUBLIC KEY-----",
    "NotBefore": "2020-10-13T03:09:00Z",
    "Arn": "acs:kms:cn-hangzhou:159498693826****:certificate/9a28de48-8d8b-484d-a766-dec4****",
    "Tags": "[{\"TagKey\":\"S1key1\",\"TagValue\":\"S1val1\"},{\"TagKey\":\"S1key2\",\"TagValue\":\"S2val2\"}]"
}
```


错误码

HttpCode	错误码	错误信息	描述
404	Certificate.NotFound	The specified certificate is not found.	指定的证书不存在。

访问错误中心查看更多错误码。

10.5. UpdateCertificateStatus

调用UpdateCertificateStatus接口更新证书状态。

本文将提供一个示例，将ID为 `9a28de48-8d8b-484d-a766-dec4****` 证书的状态更新为已禁用。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateCertificateStatus	要执行的操作，取值：UpdateCertificateStatus。
CertificateId	String	是	9a28de48-8d8b-484d-a766-dec4****	证书ID。证书管家中证书的全局唯一标识符。
Status	String	是	INACTIVE	证书的状态，取值： • INACTIVE：已禁用。 • ACTIVE：已启用。 • REOVKED：已吊销。 说明 当证书状态为REOVKED（已吊销）时，不能进行签名操作，只能进行验签操作。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
RequestId	String	e3f57fe0-9ded-40b0-9caf-a3815f2148c1	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=UpdateCertificateStatus
&CertificateId=9a28de48-8d8b-484d-a766-dec4****
&Status=INACTIVE
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <RequestId>e3f57fe0-9ded-40b0-9caf-a3815f2148c1</RequestId>
 </KMS>
```

`JSON` 格式

```
{
 "RequestId": "e3f57fe0-9ded-40b0-9caf-a3815f2148c1"
}
```

错误码

HttpCode	错误码	错误信息	描述
404	Certificate.NotFound	The specified certificate is not found.	指定的证书不存在。

访问错误中心查看更多错误码。

10.6. DeleteCertificate

调用DeleteCertificate接口删除证书及其对应的私钥和证书链。

证书及其对应的私钥和证书链删除后将无法恢复，请谨慎操作。

本文将提供一个示例，删除ID为 `9a28de48-8d8b-484d-a766-dec4****` 的证书及其对应的私钥和证书链。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteCertificate	要执行的操作，取值：DeleteCertificate。
CertificateId	String	是	9a28de48-8d8b-484d-a766-dec4****	证书ID。证书管家中证书的全局唯一标识符。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
RequestId	String	d97f6c33-ca26-4de2-a580-0e2fd1c5bfb0	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DeleteCertificate
&CertificateId=9a28de48-8d8b-484d-a766-dec4****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <RequestId>d97f6c33-ca26-4de2-a580-0e2fd1c5bfb0</RequestId>
</KMS>
```

`JSON` 格式

```
{
 "RequestId": "d97f6c33-ca26-4de2-a580-0e2fd1c5bfb0"
}
```

错误码

HttpCode	错误码	错误信息	描述
404	Certificate.NotFound	The specified certificate is not found.	指定的证书不存在。

访问错误中心查看更多错误码。

10.7. CertificatePrivateKeySign

调用CertificatePrivateKeySign接口使用指定证书生成数字签名。

请求参数中签名算法需要跟密钥类型对应。签名算法和密钥类型对照表如下：

Algorithm	Key Spec
RSA_PKCS1_SHA_256	RSA_2048
RSA_PSS_SHA_256	RSA_2048
ECDSA_SHA_256	EC_P256
SM2DSA	EC_SM2

本文将提供一个示例，使用ID为 `12345678-1234-1234-1234-12345678****` 的证书，通过 `ECDSA_SHA_256` 签名算法为原始数据 `VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4=` 生成数字签名。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CertificatePrivateKeySign	要执行的操作，取值：CertificatePrivateKeySign。
Algorithm	String	是	ECDSA_SHA_256	签名算法。取值： • RSA_PKCS1_SHA_256 • RSA_PSS_SHA_256 • ECDSA_SHA_256 • SM2DSA 说明 SM2DSA签名算法仅在中国内地使用托管密码机的地域支持。更多信息，请参见托管密码机概述。
CertificateId	String	是	12345678-1234-1234-1234-12345678****	证书ID。证书管家中证书的全局唯一标识符。
Message	String	是	VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4=	待签名数据。 使用Base64编码。例如：待签名数据的十六进制内容为 `[0x31, 0x32, 0x33, 0x34]` ，则对应的Base64编码为 `MTIzNA==` 。 当MessageType取值为RAW时，数据内容需小于4KB。 如果待签名数据内容大于4KB，您可以将MessageType指定为DIGEST，将Message指定为本地计算的消息摘要（又称哈希值）。证书管家将使用您自己的证书应用系统计算消息摘要，使用的消息摘要算法须与指定签名算法需要的消息摘要算法保持一致。具体如下： • RSA_PKCS1_SHA_256、RSA_PSS_SHA_256和ECDSA_SHA_256对应的消息摘要算法为SHA-256。 • SM2DSA对应的消息摘要算法为SM3。 说明 当证书密钥规格为EC_SM2，并且MessageType为DIGEST时，Message值为GB/T 32918.2-2016 6.1中描述的 `e` 。
MessageType	String	是	RAW	消息类型。取值： • RAW（默认值）：原始数据。 • DIGEST：原始数据的消息摘要（哈希值）。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
CertificateId	String	12345678-1234-1234-1234-12345678****	证书ID。
RequestId	String	5979d897-d69f-4fc9-87dd-f3bb73c40b80	请求ID。

名称	类型	示例值	描述
SignatureValue	String	ZOyIygCyaOW6Gj****MlNKiuyjfzw=	签名值。 使用Base64编码。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CertificatePrivateKeySign
&Algorithm=ECDSA_SHA_256
&CertificateId=12345678-1234-1234-1234-12345678****
&Message=VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4=
&MessageType=RAW
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
   <CertificateId>12345678-1234-1234-1234-123456789012</CertificateId>
   <SignatureValue>ZOyIygCyaOW6Gj****MlNKiuyjfzw=</SignatureValue>
   <RequestId>5979d897-d69f-4fc9-87dd-f3bb73c40b80</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "CertificateId": "12345678-1234-1234-1234-123456789012",
  "SignatureValue": "ZOyIygCyaOW6Gj****MlNKiuyjfzw=",
  "RequestId": "5979d897-d69f-4fc9-87dd-f3bb73c40b80"
}
```

错误码

HttpCode	错误码	错误信息	描述
404	Certificate.NotFound	The specified certificate is not found.	指定的证书不存在。

访问错误中心查看更多错误码。

10.8. CertificatePublicKeyVerify

调用CertificatePublicKeyVerify接口使用指定证书验证数字签名。

请求参数中签名算法需要跟密钥类型对应。签名算法和密钥类型对照表如下：

Algorithm	Key Spec
RSA_PKCS1_SHA_256	RSA_2048
RSA_PSS_SHA_256	RSA_2048
ECDSA_SHA_256	EC_P256
SM2DSA	EC_SM2

本文将提供一个示例，使用ID为 `12345678-1234-1234-1234-12345678****` 的证书，通过 `ECDSA_SHA_256` 签名算法验证原始数据签名 `VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4=` 的数字签名值是否为 `ZOyIygCyaOW6Gj****MlNKiuyjfzw=` 。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CertificatePublicKeyVerify	要执行的操作，取值：CertificatePublicKeyVerify。

名称	类型	是否必选	示例值	描述
Algorithm	String	是	ECDSA_SHA_256	签名算法。取值： • RSA_PKCS1_SHA_256 • RSA_PSS_SHA_256 • ECDSA_SHA_256 • SM2DSA 说明 SM2DSA签名算法仅在中国内地使用托管密码机的地域支持。更多信息，请参见托管密码机概述。
CertificateId	String	是	12345678-1234-1234-1234-12345678****	证书ID。证书管家中证书的全局唯一标识符。
Message	String	是	VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4=	原始签名数据。 使用Base64编码。例如：原始签名数据的十六进制内容为 `[0x31, 0x32, 0x33, 0x34]`，则对应的Base64编码为 `MTIzNA==`。 当MessageType取值为RAW时，数据内容需小于4KB。 如果待签名数据内容大于4KB，您可以将MessageType指定为DIGEST，将Message指定为本地计算的消息摘要（又称哈希值）。证书管家将使用您自己的证书应用系统计算消息摘要，使用的消息摘要算法须与指定签名算法需要的消息摘要算法保持一致。具体如下： • RSA_PKCS1_SHA_256、RSA_PSS_SHA_256和ECDSA_SHA_256对应的消息摘要算法为SHA-256。 • SM2DSA对应的消息摘要算法为SM3。 说明 当证书密钥规格为EC_SM2，并且MessageType为DIGEST时，Message值为GB/T 32918.2-2016 6.1中描述的 `e`。
MessageType	String	是	RAW	消息类型。取值： • RAW（默认值）：原始数据。 • DIGEST：原始数据的消息摘要（哈希值）。
SignatureValue	String	是	ZOyIygCyaOW6Gj****MlNKiuyjfzw=	签名值。 使用Base64编码。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
CertificateId	String	12345678-1234-1234-1234-12345678****	证书ID。
RequestId	String	5979d897-d69f-4fc9-87dd-f3bb73c40b80	请求ID。
SignatureValid	Boolean	true	验证结果。取值： • true：验证成功。 • false：验证失败。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CertificatePublicKeyVerify
&Algorithm=ECDSA_SHA_256
&CertificateId=12345678-1234-1234-1234-12345678****
&Message=VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4=
&MessageType=RAW
&SignatureValue=ZOyIygCyaOW6Gj****MlNKiuyjfzw=
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
    <CertificateId>12345678-1234-1234-1234-12345678****</CertificateId>
    <SignatureValid>true</SignatureValid>
    <RequestId>5979d897-d69f-4fc9-87dd-f3bb73c40b80</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "CertificateId": "12345678-1234-1234-1234-12345678****",
  "SignatureValid": "true",
  "RequestId": "5979d897-d69f-4fc9-87dd-f3bb73c40b80"
}
```

错误码

HttpCode	错误码	错误信息	描述
404	InvalidAccessKeyId.NotFound	The specified AccessKey ID does not exist.	AccessKey ID找不到。请检查调用时是否使用了正确的AccessKey。

访问错误中心查看更多错误码。

10.9. CertificatePublicKeyEncrypt

调用CertificatePublicKeyEncrypt接口使用指定证书加密数据。

使用限制：请求参数中加密算法需要跟密钥类型对应。

加密算法和密钥类型对照表如下：

Algorithm	Key Spec
RSAES_OAEP_SHA_1	RSA_2048
RSAES_OAEP_SHA_256	RSA_2048
SM2PKE	EC_SM2

本文将提供一个示例，使用ID为 `12345678-1234-1234-1234-12345678****` 的证书，通过 `RSAES_OAEP_SHA_256` 加密算法对数据 `VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4=` 进行加密。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CertificatePublicKeyEncrypt	要执行的操作，取值：CertificatePublicKeyEncrypt。
Algorithm	String	是	RSAES_OAEP_SHA_256	加密算法，取值： • RSAES_OAEP_SHA_1 • RSAES_OAEP_SHA_256 • SM2PKE 说明 SM2PKE加密算法仅在中国内地使用托管密码机的地域支持。更多信息，请参见托管密码机概述。
CertificateId	String	是	12345678-1234-1234-1234-12345678****	证书ID。证书管家中证书的全局唯一标识符。
Plaintext	String	是	VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4=	待加密数据。 使用Base64编码。例如：待加密数据的十六进制内容为 `[0x31, 0x32, 0x33, 0x34]`，则对应的Base64编码为 `MTIzNA==`。 Plaintext数据大小限制根据Algorithm的取值有所不同，具体如下： • RSAES_OAEP_SHA_1：214字节。 • RSAES_OAEP_SHA_256：190字节。 • SM2PKE：6047字节。 如果需要对超出接口限制的数据进行加密，可以先调用GenerateDataKey接口生成数据加密密钥加密数据，再调用CertificatePublicKeyEncrypt接口加密数据密钥。

名称	类型	是否必选	示例值	描述

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
CertificateId	String	12345678-1234-1234-1234-12345678****	证书ID。
CiphertextBlob	String	ZOyIygCyaOW6Gj****MlNKiuyjfzw=	加密后的密文。 使用Base64编码。
RequestId	String	5979d897-d69f-4fc9-87dd-f3bb73c40b80	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CertificatePublicKeyEncrypt
&Algorithm=RSAES_OAEP_SHA_256
&CertificateId=12345678-1234-1234-1234-12345678****
&Plaintext=VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4=
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <CertificateId>12345678-1234-1234-1234-12345678****</CertificateId>
        <CiphertextBlob>ZOyIygCyaOW6Gj****MlNKiuyjfzw=</CiphertextBlob>
        <RequestId>5979d897-d69f-4fc9-87dd-f3bb73c40b80</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "CertificateId": "12345678-1234-1234-1234-12345678****",
  "CiphertextBlob": "ZOyIygCyaOW6Gj****MlNKiuyjfzw=",
  "RequestId": "5979d897-d69f-4fc9-87dd-f3bb73c40b80"
}
```

错误码

HttpCode	错误码	错误信息	描述
404	Certificate.NotFound	The specified certificate is not found.	指定的证书不存在。

访问错误中心查看更多错误码。

10.10. CertificatePrivateKeyDecrypt

调用CertificatePrivateKeyDecrypt接口使用指定证书解密数据。

使用限制：请求参数中加密算法需要跟密钥类型对应。

加密算法和密钥类型对照表如下：

Algorithm	Key Spec
RSAES_OAEP_SHA_1	RSA_2048

Algorithm	Key Spec
RSAES_OAEP_SHA_256	RSA_2048
SM2PKE	EC_SM2

本文将提供一个示例，使用ID为 `12345678-1234-1234-1234-12345678****` 的证书，通过 `RSAES_OAEP_SHA_256` 加密算法对数据 `ZOyIygCyaOW6Gj****MlNKiuyjfzw=` 进行解密。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CertificatePrivateKeyDecrypt	要执行的操作，取值：CertificatePrivateKeyDecrypt。
Algorithm	String	是	RSAES_OAEP_SHA_256	加密算法，取值： • RSAES_OAEP_SHA_1 • RSAES_OAEP_SHA_256 • SM2PKE 说明 SM2PKE加密算法仅在中国内地使用托管密码机的地域支持。更多信息，请参见托管密码机概述。
CertificateId	String	是	12345678-1234-1234-1234-12345678****	证书ID。证书管家中证书的全局唯一标识符。
CiphertextBlob	String	是	ZOyIygCyaOW6Gj****MlNKiuyjfzw=	待解密数据。 使用Base64编码。

关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
CertificateId	String	12345678-1234-1234-1234-12345678****	证书ID。
Plaintext	String	VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4	解密后的数据。 使用Base64编码。
RequestId	String	5979d897-d69f-4fc9-87dd-f3bb73c40b80	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CertificatePrivateKeyDecrypt
&Algorithm=RSAES_OAEP_SHA_256
&CertificateId=12345678-1234-1234-1234-12345678****
&CiphertextBlob=ZOyIygCyaOW6Gj****MlNKiuyjfzw=
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <CertificateId>12345678-1234-1234-1234-12345678****</CertificateId>
        <Plaintext>VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4</Plaintext>
        <RequestId>5979d897-d69f-4fc9-87dd-f3bb73c40b80</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "CertificateId": "12345678-1234-1234-1234-12345678****",
  "Plaintext": "VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4",
  "RequestId": "5979d897-d69f-4fc9-87dd-f3bb73c40b80"
}
```

错误码

HttpCode	错误码	错误信息	描述
404	Certificate.NotFound	The specified certificate is not found.	指定的证书不存在。

访问错误中心查看更多错误码。

11.标签

11.1. TagResource

调用TagResource接口为主密钥、凭据或证书绑定标签。

使用限制：您最多可以为主密钥、凭据或证书分别绑定10个标签。

本文将提供一个示例，为ID为 `08c33a6f-4e0a-4a1b-a3fa-7ddf****` 的密钥绑定标签 `[{\"TagKey\":\"S1key1\",\"TagValue\":\"S1val1\"},{\"TagKey\":\"S1key2\",\"TagValue\":\"S2val2\"}]` 。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	TagResource	要执行的操作，取值：TagResource。
Tags	String	是	[{\"TagKey\":\"S1key1\",\"TagValue\":\"S1val1\"},{\"TagKey\":\"S1key2\",\"TagValue\":\"S2val2\"}]	一个或多个标签。格式为Tag对象数组。 Tag对象属性如下： • TagKey：标签键。 • TagValue：标签值。
KeyId	String	否	08c33a6f-4e0a-4a1b-a3fa-7ddf****	密钥ID。主密钥（CMK）的全局唯一标识符。 说明 KeyId、SecretName和CertificateId必须且只能指定其中一个参数。
SecretName	String	否	MyDbC****	凭据名称。 说明 KeyId、SecretName和CertificateId必须且只能指定其中一个参数。
CertificateId	String	否	770dbe42-e146-43d1-a55a-1355db86****	证书ID。 说明 KeyId、SecretName和CertificateId必须且只能指定其中一个参数。

关于公共请求参数的详情，请参见公共参数。

返回数据

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

示例

请求示例

```
http(s)://[Endpoint]/?Action=TagResource
&KeyId=08c33a6f-4e0a-4a1b-a3fa-7ddf****
&Tags=[{\"TagKey\":\"S1key1\",\"TagValue\":\"S1val1\"},{\"TagKey\":\"S1key2\",\"TagValue\":\"S2val2\"}]
&<公共请求参数>
```

正常返回示例

`XML` 格式

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

`JSON` 格式

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

错误码

访问错误中心查看更多错误码。

11.2. UntagResource

调用UntagResource接口为主密钥、凭据或证书解绑标签。

本文将提供一个示例，为ID为 `08c33a6f-4e0a-4a1b-a3fa-7ddf****` 的密钥解绑标签键为tagkey1、tagkey2的标签。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UntagResource	要执行的操作，取值：UntagResource。
TagKeys	String	是	["tagkey1","tagkey2"]	一个或多个标签键，多个标签键用半角逗号（,）间隔。 您只需指定标签键，不需要指定标签值。 长度为1~128个字节。
KeyId	String	否	08c33a6f-4e0a-4a1b-a3fa-7ddf****	密钥ID。主密钥（CMK）的全局唯一标识符。 说明 KeyId、SecretName和CertificateId必须且只能指定其中一个参数。
SecretName	String	否	MyDbC****	凭据名称。 说明 KeyId、SecretName和CertificateId必须且只能指定其中一个参数。
CertificateId	String	否	770dbe42-e146-43d1-a55a-1355db86****	证书ID。 说明 KeyId、SecretName和CertificateId必须且只能指定其中一个参数。

关于公共请求参数的详情，请参见公共参数。

返回数据

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

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=UntagResource
&KeyId=08c33a6f-4e0a-4a1b-a3fa-7ddf****
&TagKeys=["tagkey1","tagkey2"]
&<公共请求参数>
```

正常返回示例

`XML` 格式

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

`JSON` 格式

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

错误码

访问错误中心查看更多错误码。

11.3. ListResourceTags

调用ListResourceTags获取用户主密钥的标签。

请求格式：KeyId="string"

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListResourceTags	系统规定参数。取值：ListResourceTags。
KeyId	String	是	1234abcd-12ab-34cd-56ef-12345678****	全局唯一标识符。

返回数据

名称	类型	示例值	描述
RequestId	String	4162a6af-bc99-40b3-a552-89dcc8aaf7c8	本次请求的ID。
Tags	Array		标签。
Tag			
KeyId	String	33caea95-c3e5-4b3e-a9c6-cec76e4e****	全局唯一标识符。
TagKey	String	Project	标签键。
TagValue	String	Test	标签值。

示例

请求示例

```
https://kms.cn-hangzhou.aliyuncs.com/?Action=ListResourceTags
&KeyId=<keyid>
&<其他公共参数>
```

正常返回示例

`XML` 格式

```
<KMS>
        <RequestId>0f900dad-c747-4170-9962-1bfb6b31436b</RequestId>
        <Tags>
              <Tag>
                    <KeyId>33caea95-c3e5-4b3e-a9c6-cec76e4e****</KeyId>
                    <TagKey>Project</TagKey>
                    <TagValue>Test</TagValue>
              </Tag>
        </Tags>
</KMS>
```

`JSON` 格式

```
{
    "RequestId": "4162a6af-bc99-40b3-a552-89dcc8aaf7c8",
    "Tags": {
            "Tag": [
                {
                    "KeyId": "33caea95-c3e5-4b3e-a9c6-cec76e4e****",
                    "TagKey": "Project",
                    "TagValue": "Test"
                }
            ]
        }
}
```

错误码

HttpCode	错误码	错误信息	描述
400	Throttling	Request was denied due to request throttling.	您这个时段的流量已经超限。如果不能满足现有业务要求可以提工单进行申请。

访问错误中心查看更多错误码。

12.其他

12.1. DescribeRegions

调用DescribeRegions接口查询当前账号的可用地域列表。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeRegions	要执行的操作，取值：DescribeRegions。

返回数据

名称	类型	示例值	描述
Regions	Array of Region		地域。
Region			
RegionId	String	cn-hangzhou	地域ID。
RequestId	String	815240e2-aa37-4c26-9cca-05d4df3e8fe6	请求ID。

示例

请求示例

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

正常返回示例

`XML` 格式

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

`JSON` 格式

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

错误码

访问错误中心查看更多错误码。

12.2. OpenKmsService

调用OpenKmsService接口为当前阿里云账号开通密钥管理服务。

调用该接口前，请关注：

- 密钥管理服务需要收费，计费详情请参见计费说明。
- 一个阿里云账号只能开通一次。

- 请确保阿里云账号已通过实名认证。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OpenKmsService	要执行的操作，取值：OpenKmsService。

返回数据

名称	类型	示例值	描述
RequestId	String	3455b9b4-95c1-419d-b310-db6a53b09a39	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OpenKmsService
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<KMS>
      <RequestId>3455b9b4-95c1-419d-b310-db6a53b09a39</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "RequestId": "3455b9b4-95c1-419d-b310-db6a53b09a39"
}
```

错误码

HttpCode	错误码	错误信息	描述
400	CreateLXOrderFailed	Create order failed.	创建订单失败。
400	Forbidden.NoRealNameAuthentication	Real name authentication is needed.	未通过实名认证。
400	Forbidden.Opened	Your kms service has been opened.	您已开通密钥管理服务。

访问错误中心查看更多错误码。

12.3. DescribeAccountKmsStatus

调用DescribeAccountKmsStatus接口查询当前阿里云账号的密钥管理服务状态。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeAccountKmsStatus	要执行的操作，取值：DescribeAccountKmsStatus。

返回数据

名称	类型	示例值	描述

名称	类型	示例值	描述
AccountStatus	String	Enabled	当前阿里云账号的密钥管理服务状态，取值： • Enabled：已开通，可正常使用。 • NotEnabled：未开通。 • InDebt：已欠费，即将停止服务。 说明 当您的阿里云账号欠费后，请及时续费，以免对您的业务造成影响。 • Suspended：已停止服务。
RequestId	String	3455b9b4-95c1-419d-b310-db6a53b09a39	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DescribeAccountKmsStatus
```

正常返回示例

`XML` 格式

```
<KMS>
    <AccountStatus>Enabled</AccountStatus>
    <RequestId>3455b9b4-95c1-419d-b310-db6a53b09a39</RequestId>
</KMS>
```

`JSON` 格式

```
{
  "AccountStatus": "Enabled",
  "RequestId": "3455b9b4-95c1-419d-b310-db6a53b09a39"
}
```

错误码

访问错误中心查看更多错误码。