目录

密钥管理服务(KKMS)OpenAPI概览

API版本Version 2016-03-04

密钥管理服务(KKMS)相关接口

接口功能	Action Name	功能描述
创建主密钥	CreateKey	创建客户的主密钥
删除主密钥	DeleteKey	删除客户的主密钥；**注意**：只有禁用状态下的密钥可以被删除
修改主密钥	ModifyKey	修改用户主密钥【密钥名称】、【备注】
修改主密钥状态	ModifyKeyState	修改用户主密钥的状态，如：启用、禁用密钥
查询主密钥	DescribeKeys	查询用户主密钥
加密	Encrypt	加密
解密	Decrypt	解密
创建数据密钥	GenerateDataKey	返回一个数据加密密钥，您可以在应用程序中使用该加密密钥在本地加密数据

请求结构

客户调用金山云密钥管理服务(KKMS)的openAPI接口是通过向指定服务地址发送请求，并按照openAPI文档说明在请求中添加相应的公共参数和接口参数来完成的。

密钥管理服务(KKMS)openAPI的请求结构组成如下：

1. 服务地址

密钥管理的服务接入地址为：kkms.api.ksyun.com

2. 通信协议

支持通过 HTTP 或 HTTPS 两种方式进行请求通信，推荐使用安全性更高的 HTTPS方式发送请求。

3. 请求方法

密钥管理服务(KKMS)的openAPI同时支持GET和POST请求，推荐使用GET请求方式。

注意

- 不能混合使用两种请求方式。如果使用 GET 方式，参数均从 querystring 取得；如果使用 POST 方式，参数均从 请求Body中取得
- 如果请求方式是GET，需要对所有请求参数做URL编码；如果请求方式是POST，需要使用x-www-form-urlencoded方式进行编码。

4. 请求参数

金山云openAPI请求包含两类参数：公共请求参数和接口请求参数。其中，公共请求参数是每个接口都要用到的请求参数，具体可参见[公共参数与签名机制]https://docs.ksyun.com/documents/40377)小节；接口请求参数是各个接口所特有的，具体见各个接口的“请求参数”描述。

5. 字符编码

请求及返回结果都使用base64进行编码。

返回结果

调用金山云的openAPI服务，调用成功，返回的HTTP状态码（Status）为200；调用失败，返回4xx 或5xx的HTTP状态码（Status）。

金山云的**密钥管理(KKMS)**服务的调用返回的数据格式支持xml和json两种，默认返回xml格式，可通过设置HTTP Header Accept=application/json来改变返回数据格式。

调用成功

xml格式示例

```
<!--结果的根结点-->
<接口名称+Response>
    <ResponseMetadata>
        <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
    </ResponseMetadata>
    <!--返回结果数据-->
</接口名称+Response>
```

json格式示例

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

调用失败

调用接口失败，不会返回结果数据；HTTP请求返回一个4xx或5xx的HTTP状态码，返回的HTTP消息体中包含具体的错误代码(code)及错误信息(message)；与调用成功一样还包含请求ID(RequestId)，在调用方找不到错误原因时，可以联系金山云客服，并提供RequestId，以便我们尽快帮您解决问题。

xml格式示例

```
<!--结果的根结点-->
<ErrorResponse>
    <RequestId>e1eac1b3-1f35-44ba-abd4-7c4b7a9859f3</RequestId>
    <!--返回具体错误消息-->
    <Error>
        <!--错误来源-->
        <Type>Sender</Type>
        <!--错误代码-->
        <Code>InvalidParameterValue</Code>
        <!--错误消息-->
        <Message>An invalid or out-of-range value was supplied for the input parameter PathPrefix.</Message>
    </Error>
</ErrorResponse>
```

json格式示例

```
{
    "RequestId": "68093a99-2f63-4f39-8f70-3047ab8ecb5b",
    "Error": {
        "Type": "Sender",
        "Code": "InvalidParameterValue",
        "Message": "An invalid or out-of-range value was supplied for the input parameter PathPrefix."
    }
}
```

公共错误

错误代码(Code)	错误消息(Message)	HTTP 状态码	中文描述（语义）
MissingAuthenticationToken	Request is missing 'Host' header.	403	请求header中缺少Host
MissingAuthenticationToken	Request is missing Authentication Token.	403	请求header中缺少认证token
MissingAuthenticationToken	%s not in Http Header.	403	%s不在Http header中
SignatureDoesNotMatch	Host' must be a 'SignedHeader' in the Authorization.	403	请求的SignedHeader中必须包含Host
SignatureDoesNotMatch	Credential should be scoped with a valid terminator: 'aws4_request', not: %s.	403	请求Authorization header中的“Credential”末尾必须是“aws4_request”
SignatureDoesNotMatch	Credential should be scoped to a valid region, not:%s.	403	请求Authorization header中的“Credential”中的Region信息无效

SignatureDoesNotMatch	Credential should be scoped to correct service: %s.	403	请求Authorization header中的“Credential”中的Service信息无效
SignatureDoesNotMatch	The request signature we calculated does not match the signature you provided.	403	请求中提供的签名与实际计算结果不匹配
SignatureDoesNotMatch	Signature expired:%s.	403	签名已过期
SignatureDoesNotMatch	Date in Credential scope does not match YYYYMMDD from ISO-8601 version of date from HTTP.	403	请求Authorization header中的“Credential”中的Date应该是ISO8601基本格式，形如”YYYYMMDD“
InvalidClientTokenId	The security token included in the request is invalid.	403	请求中提供的AccessKeyId无效
AccessDenied	User: %s is not authorized to perform: %s.	403	用户%s无权限操作该资源：%s
IncompleteSignature	Date must be in ISO-8601 'basic format'. Got '%s'. See http://en.wikipedia.org/wiki/ISO_8601.	400	Date必须符合ISO_8601基本格式，参考：http://en.wikipedia.org/wiki/ISO_8601
IncompleteSignature	KSC query-string parameters must include %s. Re-examine the query-string parameters.	400	查询条件中缺少签署信息，查询条件中必须包含”X-Amz-Algorithm“、”X-Amz-Credential“、”X-Amz-SignedHeaders“、”X-Amz-Date“信息
IncompleteSignature	Unsupported ksc 'algorithm': %s.	400	只支持如下签名算法：AWS4-HMAC-SHA256
IncompleteSignature	Authorization header requires 'Credential' parameter. Authorization=%s.	400	请求Authorization header中需要包含“Credential”参数
IncompleteSignature	Credential must have exactly 5 slash-delimited elements, e.g. accesskeyid/date/region/service/aws4_request, got: %s.	400	请求Authorization header中中“Credential”至少包含5项以斜杠分隔的元素，如：keyid/date/region/service/aws4_request
IncompleteSignature	Authorization header format error.	400	请求Authorization header的格式错误
IncompleteSignature	Authorization header requires existence of either a 'X-Amz-Date' or a 'Date' header, Authorization=%s	400	请求中缺少“X-Amz-Date”或者“Date” header信息
IncompleteSignature	Authorization header requires 'Signature' parameter. Authorization=%s	400	请求Authorization header中缺少“Signature”信息
IncompleteSignature	Authorization header requires 'SignedHeaders' parameter. Authorization=%s	400	请求Authorization header中缺少“SignedHeaders”信息
ServiceUnavailable	Exception %s	500	服务暂不可用
ServiceUnavailable	Auth Service is unavailable because of an unknown error, exception or failure	500	验签或授权服务暂不可用
ServiceUnavailable	Request was rejected because it referenced an 'InnerApi' that does not have an internal service	404	请求被拒绝，因其引用的InnerAPI无内部服务。
ServiceUnavailable	OpenAPI or Service is unavailable because of an unknown error, exception or failure.	500	openAPI或服务暂不可用。
DryRunOperation	Request would have succeeded, but DryRun flag is set	412	请求本可成功，但由于设置DryRun标记未成功
NoSuchEntity	Request was rejected because it referenced an 'InnerApi' that does not exist.	404	请求被拒绝，因其引用的InnerAPI不存在
LimitExceeded	Request was rejected because the request speed of this openAPI is beyond the current flow control limit.	409	请求被拒绝，因该openAPI接口访问速度已达到流控上限
InvalidParameterValue	An invalid or out-of-range value was supplied for the input parameter %s.	400	输入参数%s的值无效、不合法或者超出范围
InvalidMethod	The method %s for is not valid for this web service.	400	Method %s对当前web服务无效
MissingParameter	An value must be supplied for the input parameter %s.	400	输入参数 %s的值不能为空
InvalidQueryParameter	The query parameter %s is malformed or does not adhere to KSC standards.	400	查询参数 %s格式不对、不存在或者不符合金山云标准

ServiceTimeout	Internal Service is unavailable because of time out.	500	内部服务由于超时暂不可用

公共参数与签名机制

金山云OpenAPI支持以下两种签名算法，您可以根据业务需要选择所使用的签名算法，请注意两种签名算法所使用的公共参数有所区别。

（1）简化版签名算法，相比AWS签名算法，签名机制更加简单。

- 公共参数
- 签名算法

（2）AWS签名算法版本4，具体可以参考AWS文档

- 公共参数
- 签名算法

数据类型

Key(客户主密钥的信息)

Contents(内容)

KeyId

- 客户主KEY的ID
- 类型:String
- 是否可缺省: 否

KeyName

- 客户主KEY的名称
- 类型:String
- 是否可缺省: 是
- 缺省值：ksc_cmk

CreateTime

- 创建时间
- 类型:String
- 是否可缺省: 否

Description

- 描述
- 类型: String
- 是否可缺省: 否

KeyState

- KEY的状态
- 类型: String
- 有效值： Enabled | Disabled | PendingDeletion | PendingImport
- 是否可缺省: 否

KeyUsage

- 客户的主KEY，仅可用于对称加密和解密
- 类型: String
- 是否可缺省: 否
- 有效值：EncryptDecrypt

Origin

- 客户的主KEY的来源
- 类型: String
- 是否可缺省: 否
- 有效值：kms | extrnal

创建主密钥

CreateKey（创建客户的主KEY）

Request Parameters（请求参数）

KeyName

- 客户主KEY的名称
- 类型:String
- 是否可缺省: 否

Description

- 备注
- 类型: String
- 是否可缺省: 是

KeyUsage

- 客户主KEY的用途，仅可用于对称加密和解密
- 类型: String
- 是否可缺省: 否
- 有效值：EncryptDecrypt

Origin

- 客户的主KEY的来源
- 类型: String
- 是否可缺省: 否
- 有效值：kms

ChargeType

- KKMS的计费类型
- 类型: String
- 有效值:
- PostPaidByDay：按日月结，无到期时间。
- 是否可缺省: 否

Response Elements（返回值）

RequestId

- 请求ID
- 类型: String

Key

- KEY的信息
- 类型: Key

Examples（举例）

Sample Request(请求)

```
http://kkms.api.ksyun.com?Action=CreateKey&Version=2016-03-04
&KeyName=test111
&KeyUsage=EncryptDecrypt&Origin=kms
&ChargeType=PostPaidByDay
```

Sample Response（返回）

```
<response>
    <RequestId>a5aaf74a-179c-4d4c-8cce-6c6b86167565</RequestId>
    <Key>
        <KeyId>0kes2r8k-vbse-87pj-icck-qu9cao8vubi7</KeyId>
        <KeyName>test111</KeyName>
        <CreateTime>2018-10-12 13:43:52</CreateTime>
        <Description></Description>
        <KeyState>Enabled</KeyState>
        <KeyUsage>EncryptDecrypt</KeyUsage>
        <Origin>kms</Origin>
    </Key>
</response>
```

删除主密钥

DeleteKey(删除用户的主Key)

Request Parameters（请求参数）

KeyId

- 客户主KEY的ID
- 类型:String
- 是否可缺省: 否

Response Elements（返回值）

RequestId

- 请求ID
- 类型: String

Return

- 操作是否成功
- 类型: Boolean

Samples（举例）

Sample Request(请求)

```
http://kkms.api.ksyun.com?Action=DeleteKey&Version=2016-03-04
&KeyId=rdmdt08v-qumh-20dd-t45k-7h8so1bqintm
```

Sample Response（返回）

```
<response>
    <RequestId>536f3952-a19d-49b2-ae5b-f57f66607053</RequestId>
    <Key>
        <KeyId>rdmdt08v-qumh-20dd-t45k-7h8so1bqintm</KeyId>
        <KeyName>1_rdmdt08v-qumh-20dd-t45k-7h8so1bqintm</KeyName>
        <CreateTime>2018-10-11 20:44:10</CreateTime>
        <Description></Description>
        <KeyState>Disabled</KeyState>
        <KeyUsage>EncryptDecrypt</KeyUsage>
        <Origin>kms</Origin>
    </Key>
</response>
```

修改主密钥

ModifyKey(修改用户主Key)

Request Parameters（请求参数）

KeyId

- 客户主KEY的ID
- 类型:String
- 是否可缺省: 否

KeyName

- 客户主KEY的名称
- 类型:String
- 是否可缺省: 是
- 缺省值：ksc_cmk

Description

- 备注
- 类型: String
- 是否可缺省: 是

Response Elements（返回值）

RequestId

- 请求ID
- 类型: String

Key

- KEY的信息
- 类型: Key

Examples（举例）

Sample Request(请求)

```
http://kkms.api.ksyun.com/?Action=ModifyKey&Version=2016-03-04
&KeyId=aj0p6hf4-mcd3-pej3-v9i9-rq85h4a2dojs
&KeyName=mosified
```

Sample Response（返回）

```
<response>
    <RequestId>3d16b7ca-4aa7-48ff-a97c-5ed929462c67</RequestId>
    <Key>
        <KeyID>aj0p6hf4-mcd3-pej3-v9i9-rq85h4a2dojs</KeyID>
        <KeyName>mosified</KeyName>
        <CreateTime>2018-10-09 20:15:39</CreateTime>
        <Description></Description>
        <KeyState>Enabled</KeyState>
        <KeyUsage>EncryptDecrypt</KeyUsage>
        <Origin>kms</Origin>
    </Key>
</response>
```

修改主密钥状态

ModifyKeyState(修改用户主Key的状态)

Request Parameters（请求参数）

KeyId

- 客户主KEY的ID
- 类型:String
- 是否可缺省: 否

KeyState

- KEY的状态

- 类型: String
- 有效值： Enabled | Disabled
- 是否可缺省: 否

Response Elements（返回值）

RequestId

- 请求ID
- 类型: String

Key

- KEY的信息
- 类型: Key

Examples（举例）

Sample Request(请求)

```
http://kkms.api.ksyun.com?Action=ModifyKeyState&Version=2016-03-04
&KeyId=aj0p6hf4-mcd3-pej3-v9i9-rq85h4a2dojs
&KeyState=Disabled
```

Sample Response（返回）

```
<response>
    <RequestId>de2417ec-1e7b-4758-bf14-03a9cb5b7f77</RequestId>
    <Key>
        <KeyId>aj0p6hf4-mcd3-pej3-v9i9-rq85h4a2dojs</KeyId>
        <KeyName>mosified</KeyName>
        <CreateTime>2018-10-09 20:15:39</CreateTime>
        <Description></Description>
        <KeyState>Disabled</KeyState>
        <KeyUsage>EncryptDecrypt</KeyUsage>
        <Origin>kms</Origin>
    </Key>
</response>
```

查询主密钥

DescribeKeys(查询用户主Key)

Request Parameters（请求参数）

KeyId.N

- 客户主KEY的ID
- 类型:String
- 是否可缺省: 是
- 缺省值：查询整个地域的KEY列表

Response Elements（返回值）

RequestId

- 请求ID
- 类型: String

KeySet

- KEY的信息
- 类型: Key List

Examples（举例）

Sample Request(请求)

```
http://kkms.api.ksyun.com?Action=DescribeKeys&Version=2016-03-04
```

Sample Response（返回）

```
<response>
    <RequestId>5b9e060f-87bb-494b-af62-f3a493033923</RequestId>
    <KeySet>
        <item>
            <KeyId>aj0p6hf4-mcd3-pej3-v9i9-rq85h4a2dojs</KeyId>
            <KeyName>test</KeyName>
            <CreateTime>2018-10-09 20:15:39</CreateTime>
            <Description></Description>
            <KeyState>Disabled</KeyState>
            <KeyUsage>EncryptDecrypt</KeyUsage>
            <Origin>kms</Origin>
        </item>
        <item>
            <KeyId>rdmdt08v-qumh-20dd-t45k-7h8so1bqintm</KeyId>
            <KeyName>1</KeyName>
            <CreateTime>2018-10-11 20:44:10</CreateTime>
            <Description></Description>
            <KeyState>Enabled</KeyState>
            <KeyUsage>EncryptDecrypt</KeyUsage>
            <Origin>kms</Origin>
        </item>
    </KeySet>
</response>
```

加密

Encrypt(加密)

使用客户主密钥（CMK）将明文加密成密文

Request Parameters（请求参数）

KeyId

- 客户主KEY的ID
- 类型:String
- 是否可缺省: 否

Plaintext

- 明文数据，最多长度不超过4096
- 类型:Base64-encoded binary data object
- 是否可缺省: 否

Response Elements（返回值）

RequestId

- 请求ID
- 类型: String

CiphertextBlob

- 密文数据
- 类型:Base64-encoded binary data object
- 是否可缺省: 否

KeyId

- 客户主KEY的ID
- 类型:String
- 是否可缺省: 否

Examples（举例）

Sample Request(请求)

```
http://kkms.api.ksyun.com?Action=Encrypt&Version=2016-03-04
&KeyId=aj0p6hf4-mcd3-pej3-v9i9-rq85h4a2dojs
&Plaintext=aaaa
```

Sample Response（返回）

```
<response>
    <RequestId>dff2ff24-555f-4209-aa2a-c19b0f92e551</RequestId>
    <CiphertextBlob>/KhaumND4qw/V/LAfk04aQ==</CiphertextBlob>
    <KeyId>aj0p6hf4-mcd3-pej3-v9i9-rq85h4a2dojs</KeyId>
</response>
```

解密

Decrypt(解密)

解密密文

Request Parameters（请求参数）

KeyId

- 客户主KEY的ID
- 类型:String
- 是否可缺省: 否

CiphertextBlob

- 密文数据
- 类型:Base64-encoded binary data object
- 是否可缺省: 否

Response Elements（返回值）

RequestId

- 请求ID
- 类型: String

KeyId

- 客户主KEY的ID
- 类型:String
- 是否可缺省: 否

Plaintext

- 明文数据，最多长度不超过4096
- 类型:Base64-encoded binary data object
- 是否可缺省: 否

Examples（举例）

Sample Request(请求)

```
http://kkms.api.ksyun.com?Action=Decrypt&Version=2016-03-04
&KeyId=aj0p6hf4-mcd3-pej3-v9i9-rq85h4a2dojs
&CiphertextBlob=/KhaumND4qw/V/LAfk04aQ==
```

Sample Response（返回）

```
<response>
    <RequestId>f30e5dde-d1dc-4fbb-87f0-1a81b6bce454</RequestId>
    <Plaintext>aaaa</Plaintext>
    <KeyId>aj0p6hf4-mcd3-pej3-v9i9-rq85h4a2dojs</KeyId>
</response>
```

创建数据密钥

GenerateDataKey(创建数据密钥)

Request Parameters（请求参数）

KeyId

- 客户主KEY的ID
- 类型:String
- 是否可缺省: 否

KeySpec

- 数据加密密钥（DataKey）的长度。使用AES128生成128位对称密钥，或AES256生成256位对称密钥
- 类型:String
- 有效值：AES256 | AES128
- 是否可缺省: 否

NumberOfBytes

- DataKey的长度为字节。例如，使用值64生成512位。数据键（64字节为512位）。对于公共密钥长度（128位和256位对称密钥），我们建议您使用KEYSPEC字段，而不是使用此键字段。
- 类型:Integer
- 可取值：1-1024
- 是否可缺省: 否

Response Elements（返回值）

RequestId

- 请求ID
- 类型: String

KeyId

- 客户主KEY的ID
- 类型:String
- 是否可缺省: 否

Plaintext

- DataKey的明文数据，最多长度不超过4096
- 类型:Base64-encoded binary data object
- 是否可缺省: 否

CiphertextBlob

- DataKey加密后的密文数据
- 类型:Base64-encoded binary data object
- 是否可缺省: 否

Examples（举例）

Sample Request(请求)

```
http://kkms.api.ksyun.com?Action=GenerateDataKey&Version=2016-03-04
&KeyId=aj0p6hf4-mcd3-pej3-v9i9-rq85h4a2dojs
&KeySpec=AES128&NumberOfBytes=64
```

Sample Response（返回）

```
<response>
    <RequestId>e43b83dd-a976-46c0-88c2-b33a81a995f1</RequestId>
    <KeyId>aj0p6hf4-mcd3-pej3-v9i9-rq85h4a2dojs</KeyId>
    <Plantext>EwosoIAlYOaMcfn76bl6kw==</Plantext>
    <CiphertextBlob>OwSpTUgACpNvuEEZr9ANSQ==</CiphertextBlob>
```

</response>