

阿里云

数据湖分析 API参考

文档版本：20220630

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 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.API概览	06
2.请求结构	08
3.签名方法	09
4.公共参数	11
5.返回结果	13
6.Serverless Spark	14
6.1. SubmitSparkSQL	14
6.2. SubmitSparkJob	15
6.3. GetJobStatus	17
6.4. GetJobDetail	18
6.5. GetJobLog	23
6.6. GetJobAttemptLog	24
6.7. KillSparkJob	26
6.8. ListSparkJob	27
6.9. ListSparkJobAttempt	32
6.10. ListSparkStatements	35
6.11. GetSparkStatement	38
6.12. ExecuteSparkStatement	40
6.13. CancelSparkStatement	42
6.14. GetSparkSessionState	43
7.服务管理	45
7.1. CreateInstance	45
7.2. ReleaseInstance	47
7.3. ValidateVirtualClusterName	48
8.元数据	51
8.1. 权限管理	51

8.1.1. GrantPrivileges	51
8.1.2. RevokePrivileges	53
8.2. 数据库管理	55
8.2.1. CreateDatabase	55
8.2.2. GetDatabase	57
8.2.3. GetAllDatabases	59
8.2.4. AlterDatabase	61
8.2.5. DropDatabase	63
8.3. 分区管理	65
8.3.1. AddPartitions	65
8.3.2. GetPartition	68
8.3.3. GetPartitions	72
8.3.4. DropPartition	77
8.4. 表管理	79
8.4.1. CreateTable	79
8.4.2. GetTable	82
8.4.3. GetAllTables	85
8.4.4. AlterTable	87
8.4.5. DropTable	89

1.API概览

云原生数据湖分析DLA提供以下相关API接口。

Serverless Spark

API	描述
SubmitSparkSQL	提交SparkSQL作业。
SubmitSparkJob	提交Spark作业。
GetJobStatus	获取Spark作业的执行状态。

服务管理

API	描述
CreateInstance	创建实例。
ReleaseInstance	释放实例。
ValidateVirtualClusterName	校验虚拟集群的名字是否合法。

元数据

 **说明** 元数据相关的API信息比较复杂，不建议使用。如果您有必须使用的需求可以搜索钉钉群号33899370加入钉钉群进行咨询沟通。

权限管理

API	描述
GrantPrivileges	对用户进行授权。
RevokePrivileges	移除某个用户的某个权限。

数据库管理

API	描述
CreateDatabase	创建数据库。
GetDatabase	获取给定数据库的信息。
GetAllDatabases	获取当前用户有读权限或写权限的所有数据库。
AlterDatabase	修改数据库的信息。
DropDatabase	删除数据库。

分区管理

API	描述
AddPartitions	在元信息系统中添加分区信息。
GetPartition	获取分区信息。
GetPartitions	获取表的所有分区信息。
DropPartition	删除分区信息。

表管理

API	描述
CreateTable	创建OSS的表。
GetTable	获取给定表的详细信息。
GetAllTables	获取当前用户有读权限或写权限的所有表。
AlterTable	修改表的数据库名、表名、表属性以及列信息。
DropTable	删除给定表。

2. 请求结构

获取用户AccessKey

使用阿里云API需要获取用户的AccessKey以对入参进行签名, 获取用户AccessKey的方法请参见[获取AccessKey](#)。

获取运行区的RegionId

需要确定使用区的RegionId, 每个区域的值请参见[地域和可用区](#)。

请求地址的构成

数据湖分析通过API操作Spark作业的请求结构如下所示:

```
http://openanalytics.[region-id].aliyuncs.com/?Action=XXX
&<公共请求参数>
&<接口需求参数>
```

- 所有接口支持GET方法来调用。
- 调用地址需要填入RegionId
- 公共请求参数是所有的API调用都需要带的入参, 详细说明请参见[公共参数](#)。
- 接口需求参数是每个API需要的特定输入, 参数说明可参见每个API的详情说明。

以下示例通过API来获取一个Spark作业的状态。

```
http://openanalytics-open.cn-hangzhou.aliyuncs.com/?Action=GetJobStatus
&Format=JSON
&AccessKeyId=XXX
&SignatureNonce=YYY
&Signature=ZZZ
&SignatureMethod=HMAC-SHA1
&SignatureVersion=1.0
&Version=2018-06-19
&Timestamp=2020-10-27T00:00:00Z
&JobId=MySparkJobId
&VcName=MyCluster
```

3. 签名方法

对于每一次HTTP或者HTTPS协议请求，我们会根据访问中的签名信息验证访问请求者身份。具体由使用AccessKeyId和AccessKeySecret对称加密验证实现。

 **注意** 使用SDK提交请求可以免去做签名。

步骤一：构造规范化请求字符串

1. 排序参数。排序规则以首字母顺序排序，排序参数包括公共请求参数和接口自定义参数，不包括公共请求参数中的Signature参数。使用GET方法提交请求时，这些参数就是请求URL中的参数部分，即URL中？之后由&连接的部分。
2. 编码参数。使用UTF-8字符集按照RFC3986规则编码请求参数和参数取值，编码规则如下：
 - 字符A~Z、a~z、0~9以及字符 - 、 _ 、 . 、 ~ 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。示例：半角双引号 (") 对应 %22 。
 - 扩展的UTF-8字符，编码成 %XY%ZA... 的格式。
 - 空格 () 编码成 %20 ，而不是加号 (+) 。该编码方式与 application/x-www-form-urlencodedMIME 格式编码算法相似，但又有所不同。

如果您使用的是Java标准库中的java.net.URLEncoder，可以先用标准库中percentEncode编码，随后将编码后的字符中加号(+)替换为%20、星号(*)替换为%2A、%7E替换为波浪号(~)，即可得到上述规则描述的编码字符串。

```
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;
}
```

3. 使用等号 (=) 连接编码后的请求参数和参数取值。
4. 使用与号 (&) 连接编码后的请求参数。

完成上述操作后，您就可以获取到规范化请求字符串（CanonicalizedQueryString）。

步骤二：构造签名字符串

构造待签名字符串 StringToSign 。您可以同样使用 percentEncode 处理上一步构造的规范化请求字符串，规则如下：

```
StringToSign=
    HTTPMethod + "&" + //HTTPMethod: 发送请求的 HTTP 方法，例如 GET。
    percentEncode("/") + "&" + //percentEncode("/")：字符 (/) UTF-8 编码得到的值，即 %2F。
    percentEncode(CanonicalizedQueryString) //您的规范化请求字符串。
```

按照RFC2104的定义，计算待签名字符串 StringToSign的HMAC-SHA1 值。示例使用的是Java Base64编码方法。

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

计算签名时，RFC2104规定的Key值是您的 `AccessKeySecret` 并加上与号 (`&`)，其ASCII值为38。将计算结果添加根据RFC3986 Signature规则编码后的参数到规范化请求字符串URL中。

签名计算过程示例

以调用GetJobStatus查询作业状态为例。假设您获得了 `AccessKeyId=xxx` 以及 `AccessKeySecret=yyy`，签名流程如下所示：

1. 构造规范化请求字符串，需要注意这里有一个随机字符串 `SignatureNonce`

```
http://openanalytics.cn-hangzhou.aliyuncs.com/?
AccessKeyId=xxx&Action=GetJobStatus&Format=JSON&JobId=MySparkJobId&SignatureMethod=HMAC-
SHA1&SignatureNonce=f87701c37ad49e3153fabf78ed2ad73c&SignatureVersion=1.0&Timestamp=202
0-10-27T07:32:05Z&VcName=MyCluster&Version=2018-06-19
```

2. 构造待签名字符串。

```
GET&%2F&AccessKeyId=xxx&Action=GetJobStatus&Format=JSON&JobId=MySparkJobId&SignatureMet
hod=HMAC-
SHA1&SignatureNonce=f87701c37ad49e3153fabf78ed2ad73c&SignatureVersion=1.0&Timestamp=202
0-10-
27T07%3A32%3A05Z&VcName=MyCluster&Version=2018-06-19
```

3. 计算签名值。因为 `AccessKeySecret=yyy`，用于计算的Key为 `yyy&`，计算得到的签名值为 `DR5p4dbFur6adTbYPIq8uH4sW6w=`。示例使用的是JavaBase64编码方法。

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

4. 添加RFC3986规则编码后的 `Signature=DR5p4dbFur6adTbYPIq8uH4sW6w%3D` 到URL中。

```
http://openanalytics.cn-hangzhou.aliyuncs.com/?
AccessKeyId=xxx&Action=GetJobStatus&Format=JSON&JobId=MySparkJobId&SignatureMethod=HMAC-
SHA1&SignatureNonce=f87701c37ad49e3153fabf78ed2ad73c&SignatureVersion=1.0&Timestamp=202
0-10-27T07%3A32%3A05Z&VcName=MyCluster&Version=2018-06-19&Signature=DR5p4dbFur6adTbYPIq8uH4sW6w%3D
```

通过以上URL，您可以使用浏览器、curl或者wget等工具发起HTTP请求调用。

4. 公共参数

公共参数指调用任何API时必须携带的参数，其中签名的方法请参见[签名方法](#)

 **注意** 如果使用SDK开发，公共参数可以由SDK自动生成，因此推荐使用SDK调用API。

公共请求参数

名称	类型	是否必须	描述
Action	String	是	API的名称。取值请参见 API概览
Version	String	是	API版本号，取值：2018-06-19
AccessKeyId	String	是	用户AccessKeyId
Signature	String	是	签名结果串。
SignatureMethod	string	是	签名方式。取值：HMAC-SHA1
Timestamp	String	是	请求的时间戳。日期格式按照ISO8601标准表示，并需要使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。例如，2020-10-01T12:00:00Z，表示北京时间2020年10月01日20点0分0秒。
SignatureVersion	String	是	签名算法版本。取值：1.0
SignatureNonce	String	是	唯一随机数，用于防止网络重放攻击。用户在不同请求间要使用不同的随机数值。
Format	String	是	返回参数的类型，取值：json

请求示例

以下示例查询杭州虚拟集群 `MyCluster` 下，ID为MySparkId的作业的状态：

```
http://openanalytics-open.cn-hangzhou.aliyuncs.com/?Action=GetJobStatus
    &Format=JSON
    &AccessKeyId=XXX
    &SignatureNonce=YYY
    &Signature=ZZZ
    &SignatureMethod=HMAC-SHA1
    &SignatureVersion=1.0
    &Version=2018-06-19
    &Timestamp=2020-10-27T00:00:00Z
    &JobId=MySparkJobId
    &VcName=MyCluster
```

公共返回参数

名称	类型	描述
RequestId	String	本次请求的唯一ID，可以用来记录调用详情以及追踪问题。

5. 返回结果

正常返回结果

接口调用成功后会返回接口返回参数和请求 ID，我们称这样的返回为正常返回。HTTP 状态码为 2xx。返回 JSON 结构如下所示：

```
{
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216" /* 返回结果数据 */,
  "其它可选返回": "返回值" /* 每个API自有的返回值 */
}
```

异常返回结果

接口调用出错后，会返回错误码、错误信息和请求 ID，我们称这样的返回为异常返回。HTTP 状态码为 4xx 或者 5xx。

```
{
  "RequestId": "540CFF28-407A-40B5-B6A5-74Bxxxxxxxxx", /* 请求 ID */
  "Code": "MissingParameter.CommandId", /* 错误码 */
  "Message": "Missing Param" /* 错误信息 */
}
```

6. Serverless Spark

6.1. SubmitSparkSQL

调用SubmitSparkSQL提交SparkSQL作业。

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	SubmitSparkSQL	系统规定参数。取值：SubmitSparkSQL。
Sql	String	是	set spark.driver.resou rceSpec=medium ;set spark.executor.in stances=5; set spark.executor.re sourceSpec=medi um; set spark.app.name= sparksqtest; show databases;	Spark SQL内容。具体请参考 Spark SQL 。
VcName	String	是	MySparkCluster	执行SparkSQL作业的虚拟集群名称。

返回数据

名称	类型	示例值	描述
JobId	String	j202104261729hang zhou224ee52300009 30	Spark作业ID。
RequestId	String	B570310A-8A49- 461B-B81D- AE0B58D63DA4	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=SubmitSparkSQL
&Sql=set spark.driver.resourceSpec=medium;set spark.executor.instances=5; set spark.executor.resourceSpec=medium; set spark.app.name=sparksqltest; show databases;
&VcName=MySparkCluster
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>B570310A-8A49-461B-B81D-AE0B58D63DA4</RequestId>
<JobId>j202104261729hangzhou224ee5230000930</JobId>
```

JSON 格式

```
{"RequestId":"B570310A-8A49-461B-B81D-AE0B58D63DA4","JobId":"j202104261729hangzhou224ee5230000930"}
```

错误码

访问[错误中心](#)查看更多错误码。

6.2. SubmitSparkJob

提交Spark作业。

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	SubmitSparkJob	系统规定参数。取值：SubmitSparkJob。

名称	类型	是否必选	示例值	描述
ConfigJson	String	是	{ "name": "SparkPi", "file": "local:///tmp/spark-examples.jar", "className": "org.apache.spark.examples.SparkPi", "args": ["100"], "conf": { "spark.driver.resourceSpec": "medium", "spark.executor.instances": 5, "spark.executor.resourceSpec": "medium" } }	定义Spark作业的JSON字符串。JSON字符串的作业参数说明请参见 作业配置指南 。
VcName	String	是	MyCluster	执行作业的虚拟集群名称。

返回数据

名称	类型	示例值	描述
JobId	String	j202011032013hangzhouf742a4330003679	Spark作业的ID。
RequestId	String	C039FD4D-2F3C-4556-AF09-864D3A6485B2	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=SubmitSparkJob
&ConfigJson={ "name": "SparkPi", "file": "local:///tmp/spark-examples.jar", "className": "org.apache.spark.examples.SparkPi", "args": [ "100" ], "conf": { "spark.driver.resourceSpec": "medium", "spark.executor.instances": 5, "spark.executor.resourceSpec": "medium" } }
&VcName=MyCluster
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>C039FD4D-2F3C-4556-AF09-864D3A6485B2</RequestId>
<JobId>j202011032013hangzhouf742a4330003679</JobId>
```

JSON 格式

```
{
  "RequestId": "C039FD4D-2F3C-4556-AF09-864D3A6485B2",
  "JobId": "j202011032013hangzhouf742a4330003679"
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.3. GetJobStatus

获取Spark作业的执行状态。

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	GetJobStatus	系统规定参数。取值：GetJobStatus。
JobId	String	是	j202011031935hangzhouf742a4330003667	Spark作业的ID。您可以登录 DLA控制台 ，在作业任务列表中获取作业任务ID。
VcName	String	是	MyCluster	执行作业的虚拟集群名称。

返回数据

名称	类型	示例值	描述
RequestId	String	5C3F9ADD-2350-4C21-A3A1-A906F8DF8016	请求ID。
Status	String	success	作业执行状态码。

作业的状态码如下表

状态码	语义
starting	启动中，作业尚未执行。
running	作业在执行中。
error	作业中的程序执行失败，抛出Exception。
dead	作业因为资源问题等执行失败。
killed	执行中的作业被取消。
success	作业执行成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetJobStatus
&JobId=j202011031935hangzhouf742a4330003667
&VcName=MyCluster
&<公共请求参数>
```

正常返回示例

XML 格式

```
<Status>success</Status>
<RequestId>5C3F9ADD-2350-4C21-A3A1-A906F8DF8016</RequestId>
```

JSON 格式

```
{
  "Status": "success",
  "RequestId": "5C3F9ADD-2350-4C21-A3A1-A906F8DF8016"
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.4. GetJobDetail

调用GetJobDetail获取Spark作业的详细信息。

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	GetJobDetail	系统规定参数。取值：GetJobDetail。
JobId	String	是	j202010271622hangzhouf742a4330000923	Spark作业ID。
VcName	String	是	MyCluster	执行Spark作业的虚拟集群名称。

返回数据

名称	类型	示例值	描述
JobDetail	Struct		Spark作业的详细信息。
CreateTime	String	2020-10-27 16:23:16	Spark作业开始执行的时间，时区为执行区域的时区。
CreateTimeValue	String	1603786996000	Spark作业开始执行的毫秒级时间戳。
Detail	String	{\"args\": [\"100\"],\"name\": \"SparkPi\", \"className\": \"org.apache.spark.examples.SparkPi\", \"conf\": {\"spark.driver.resourceSpec\": \"medium\", \"spark.executor.instances\": 5, \"spark.executor.resourceSpec\": \"medium\"}, \"file\": \"local:///tmp/spark-examples.jar\"}	定义Spark作业的JSON字符串。

名称	类型	示例值	描述
DriverResourceSpec	String	small	Driver计算资源规格。取值范围： small：1Core 4GB medium：2Core 8GB large：4Core 16GB xlarge：8Core 32GB
ExecutorInstances	String	1	Spark作业Executor数量。
ExecutorResourceSpec	String	small	Executor计算资源规格。取值范围： small：1Core 4GB medium：2Core 8GB large：4Core 16GB xlarge：8Core 32GB
JobId	String	j202010271622hangzhouf742a4330000923	Spark作业ID。
JobName	String	SparkPi	Spark作业的名称。
LastJobAttemptId	String	202105251618hzslo t9906b0b40000005-0001	最近一次尝试执行的Spark作业ID。
SparkUI	String	https://dlai-cn-hangzhou.aliyuncs.com/?token=xxx	Spark作业的UI，具体请参考 配置Spark UI 。
Status	String	success	Spark作业的状态码。
SubmitTime	String	2020-10-27 16:23:16	提交Spark作业的时间，时区为执行区域的时区。
SubmitTimeValue	String	1603786996000	提交Spark作业的毫秒级时间戳。
UpdateTime	String	2020-10-27 16:23:16	最后一次更新Spark作业状态的时间，时区为执行作业区域的时区。
UpdateTimeValue	String	1603786996000	最后一次更新Spark作业状态的毫秒级时间戳。

名称	类型	示例值	描述
VcName	String	MyCluster	执行Spark作业的虚拟集群名称。
RequestId	String	5F10AB6E-8984-4E32-B821-4E1512711B8C	请求的唯一ID。

作业的状态码如下表

状态码	语义
starting	启动中，作业尚未执行。
running	作业在执行中。
error	作业中的程序执行失败，抛出了Exception。
dead	作业因为资源不足等问题失败。
killed	作业在执行中被取消。
success	作业执行成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetJobDetail
&JobId=j202010271622hangzhouf742a4330000923
&VcName=MyCluster
&<公共请求参数>
```

正常返回示例

XML 格式

```
<JobDetail>
  <Status>success</Status>
  <VcName>MyCluser</VcName>
  <ExecutorInstances>5</ExecutorInstances>
  <LastJobAttemptId>j202010271622hangzhouf742a4330000923-0001</LastJobAttemptId>
  <SparkUI>https://dlaui-cn-hangzhou.aliyuncs.com/?token=xxx</SparkUI>
  <DriverResourceSpec>medium</DriverResourceSpec>
  <CreateTime>2020-10-27 16:23:16</CreateTime>
  <JobName>SparkPi</JobName>
  <ExecutorResourceSpec>medium</ExecutorResourceSpec>
  <SubmitTime>2020-10-27 16:22:52</SubmitTime>
  <CreateTimeValue>1603786996000</CreateTimeValue>
  <UpdateTimeValue>1603787047000</UpdateTimeValue>
  <SubmitTimeValue>1603786972000</SubmitTimeValue>
  <UpdateTime>2020-10-27 16:24:07</UpdateTime>
  <JobId>j202010271622hangzhouf742a4330000923</JobId>
  <Detail>{"args":["100"],"name":"SparkPi","className":"org.apache.spark.examples.SparkPi",
  "conf":{"spark.driver.resourceSpec":"medium","spark.executor.instances":5,"spark.executor.resourceSpec":"medium"},"file":"local:///tmp/spark-examples.jar"}</Detail>
</JobDetail>
<RequestId>5F10AB6E-8984-4E32-B821-4E1512711B8C</RequestId>
```

JSON 格式

```
{
  "JobDetail": {
    "Status": "success",
    "VcName": "MyCluser",
    "ExecutorInstances": 5,
    "LastJobAttemptId": "j202010271622hangzhouf742a4330000923-0001",
    "SparkUI": "https://dlaui-cn-hangzhou.aliyuncs.com/?token=xxx",
    "DriverResourceSpec": "medium",
    "CreateTime": "2020-10-27 16:23:16",
    "JobName": "SparkPi",
    "ExecutorResourceSpec": "medium",
    "SubmitTime": "2020-10-27 16:22:52",
    "CreateTimeValue": 1603786996000,
    "UpdateTimeValue": 1603787047000,
    "SubmitTimeValue": 1603786972000,
    "UpdateTime": "2020-10-27 16:24:07",
    "JobId": "j202010271622hangzhouf742a4330000923",
    "Detail": "{\"args\":[\"100\"],\"name\":\"SparkPi\",\"className\":\"org.apache.spark.examples.SparkPi\",\"conf\":{\"spark.driver.resourceSpec\":\"medium\",\"spark.executor.instances\":5,\"spark.executor.resourceSpec\":\"medium\"},\"file\":\"local:///tmp/spark-examples.jar\"}"
  },
  "RequestId": "5F10AB6E-8984-4E32-B821-4E1512711B8C"
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.5. GetJobLog

获取Spark作业的日志。

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	GetJobLog	系统规定参数。取值：GetJobLog。
JobId	String	是	j202010271622hangzhouf742a4330000923	Spark作业的ID。您可以登录DLA控制台，在作业任务列表中获取作业任务ID。
VcName	String	是	MyCluster	执行作业的虚拟集群名称。

返回数据

名称	类型	示例值	描述
Data	String	INFO SparkContext: Running Spark version 2.x	作业的日志内容。
RequestId	String	EC031B4B-3E07-40D2-9BC5-xxxx	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetJobLog
&JobId=j202010271622hangzhouf742a4330000923
&VcName=MyCluster
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>3970ACB5-3E30-4A3A-B63A-xxxx</RequestId>
<Data>Running Spark version 2.x</Data>
```

JSON 格式

```
{
  "RequestId": "3970ACB5-3E30-4A3A-B63A-xxxx",
  "Data": "Running Spark version 2.x"
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.6. GetJobAttemptLog

调用GetJobAttemptLog，获取对应JobAttempt的日志。

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	GetJobAttemptLog	系统规定参数。取值：GetJobAttemptLog。
JobAttemptId	String	是	j202105272322hangzhou5d64f1560000128-0001	Spark作业尝试ID。
JobId	String	是	j202105272322hangzhou5d64f1560000128	Spark作业ID。您可以在作业管理页面查看作业。
VcName	String	是	release-test	执行Spark作业的虚拟集群名称。您可以在虚拟集群管理页面查看虚拟集群名称。

返回数据

名称	类型	示例值	描述
----	----	-----	----

名称	类型	示例值	描述
Data	String	<pre> local:///opt/spark/jars/offline-sql.jar, main_file\n+ exec /usr/bin/tini -s -- /jdk/jdk8/bin/java - cp '/opt/tools/exec- wrapper.jar:::/opt /spark/jars/*' com.aliyun.dla.spark .SparkJobWrapper /opt/spark/bin/spar k-submit --conf spark.driver.host=17 2.16.6.205 --conf spark.ui.port=4040 - -conf 'spark.driver.extraja vaOptions=- Dlog4j.configuration =file:///opt/spark/l og- conf/log4j.propertie s - XX:+HeapDumpOnO utOfMemoryError - XX:HeapDumpPath= /tmp/dump.hprof - XX:OnOutOfMemory Error=\"bash /opt/tools/oss- cp.sh /tmp/dump.hprof oss://dla-test-cn- hangzhou/spark- logs/release- test/j202105272322 hangzhou5d64f1560 000128- 0001/driver/dump.h prof; bash /opt/tools/job- stop.sh\" ' </pre>	返回的日志。
RequestId	String	9CE8F271-F918-43B6-8F58-F9F1C2DCFD88	请求的唯一ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetJobAttemptLog
&JobAttemptId=j202105272322hangzhou5d64f1560000128-0001
&JobId=j202105272322hangzhou5d64f1560000128
&VcName=release-test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>9CE8F271-F918-43B6-8F58-F9F1C2DCFDB8</RequestId>
<Data>local:///opt/spark/jars/offline-sql.jar, main_file\n+ exec /usr/bin/tini -s -- /jdk/j
dk8/bin/java -cp '/opt/tools/exec-wrapper.jar:../opt/spark/jars/*' com.aliyun.dla.spark.S
parkJobWrapper /opt/spark/bin/spark-submit --conf spark.driver.host=172.16.6.205 --conf spa
rk.ui.port=4040 --conf 'spark.driver.extraJavaOptions=-Dlog4j.configuration=file:///opt/spa
rk/log-conf/log4j.properties -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp/dump.hpr
of -XX:OnOutOfMemoryError=\"bash /opt/tools/oss-cp.sh /tmp/dump.hprof oss://dla-test-cn-han
gzhou/spark-logs/release-test/j202105272322hangzhou5d64f1560000128-0001/driver/dump.hprof;
bash /opt/tools/job-stop.sh\" '</Data>
```

JSON 格式

```
{\"RequestId\":\"9CE8F271-F918-43B6-8F58-F9F1C2DCFDB8\",\"Data\":\"local:///opt/spark/jars/offline
-sql.jar, main_file\\n+ exec /usr/bin/tini -s -- /jdk/jdk8/bin/java -cp '/opt/tools/exec-wr
apper.jar:../opt/spark/jars/*' com.aliyun.dla.spark.SparkJobWrapper /opt/spark/bin/spark-
submit --conf spark.driver.host=172.16.6.205 --conf spark.ui.port=4040 --conf 'spark.driver
.extraJavaOptions=-Dlog4j.configuration=file:///opt/spark/log-conf/log4j.properties -XX:+He
apDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp/dump.hprof -XX:OnOutOfMemoryError=\\\"bash /
opt/tools/oss-cp.sh /tmp/dump.hprof oss://dla-test-cn-hangzhou/spark-logs/release-test/j202
105272322hangzhou5d64f1560000128-0001/driver/dump.hprof; bash /opt/tools/job-stop.sh\\\" '\"
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.7. KillSparkJob

调用KillSparkJob强行停止一个运行中的Spark作业。

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	KillSparkJob	系统规定参数。取值：KillSparkJob。

名称	类型	是否必选	示例值	描述
JobId	String	是	j202011031935hangzhouf742a4330003667	Spark作业的ID。
VcName	String	是	MyCluster	执行Spark作业的虚拟集群名称。

返回数据

名称	类型	示例值	描述
Data	String	deleted	执行结果， <code>deleted</code> 代表执行成功。
RequestId	String	3970ACB5-3E30-4A3A-B63A-1790FCC261F4	请求的唯一ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=KillSparkJob
&JobId=j202011031935hangzhouf742a4330003667
&VcName=MyCluster
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>3970ACB5-3E30-4A3A-B63A-1790FCC261F4</RequestId>
<Data>deleted</Data>
```

JSON 格式

```
{
  "RequestId": "3970ACB5-3E30-4A3A-B63A-1790FCC261F4",
  "Data": "deleted"
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.8. ListSparkJob

调用ListSparkJob用于翻页提取某个数据湖分析Spark虚拟集群的历史作业详情信息。

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	ListSparkJob	系统规定参数。取值：ListSparkJob。
PageNumber	Integer	是	1	指定需要查询的页码号，用于遍历全部数据，页码的编码从1开始。
PageSize	Integer	是	10	每页返回的作业的个数。
VcName	String	是	MySparkCluster	执行Spark作业的虚拟集群名称。
Condition	Json	否	{ "createTime": "2020-12-28 09:00:00", "jobName": "SparkPi" }	提取Spark虚拟集群的历史作业详情信息的过滤条件，不填写表示没有过滤条件。

Condition是一个JSON格式的字符串，目前支持如下参数。

- status：Spark作业的状态，取值如下。
 - starting：提交中。
 - running：运行中。
 - success：成功。
 - error：运行失败。
 - killed：被取消。
 - unknown：未知状态。
 - dead：启动失败。
- createTime：代表筛选指定时间之后提交的作业，格式为yyyy-mm-dd HH:mm:ss，指定该参数的时候，同时需要指定timeZone。
- timeZone：createTime时间串所在的时区，例如 Asia/Shanghai。
- jobId：Spark作业ID。
- jobName：Spark作业的名称，支持局部匹配。

返回数据

名称	类型	示例值	描述
----	----	-----	----

名称	类型	示例值	描述
RequestId	String	E18B0634-F003-486A-90B9-4AA6932DA3AA	本次请求的唯一编码。
DataResult	Struct		虚拟集群的历史作业结果列表。
JobList	Array of Data		历史作业详情数组。
CreateTime	String	2020-10-27 17:51:05	创建Spark作业的时间点，时区为该集群所在的地域的时区。
CreateTimeValue	String	1603792267000	创建Spark作业的毫秒级时间戳。
Detail	String	{"args":["100"],"name":...}	描述此作业的JSON格式的字符串。
DriverResourceSpec	String	small	作业所申请的Driver的实例规格。
ExecutorInstances	String	100	作业所申请的Executor的数量。
ExecutorResourceSpec	String	small	作业所申请的Executor的规格。
JobId	String	j202010271750hangzhouf742a4330000950	Spark作业ID。
JobName	String	SparkPi	Spark作业的名称。
SparkUI	String	https://dlaui-cn-hangzhou.aliyuncs.com/?token=xxx	Spark作业的UI，具体请参考文档 Spark UI 。
Status	String	running	作业的状态码。
SubmitTime	String	2020-10-27 17:51:05	提交作业的时间，时区为该集群所在的地域的时区。
SubmitTimeValue	String	1603792267000	提交作业的毫秒级时间戳。
UpdateTime	String	2020-10-27 17:51:05	最后一次更新作业状态的时间，时区为该集群所在的地域的时区。

名称	类型	示例值	描述
UpdateTimeValue	String	1603792267000	最后一次更新作业状态的毫秒级时间戳。
VcName	String	MyCluster	执行作业的虚拟集群名称。
PageNumber	String	1	当前请求返回的页码。
PageSize	String	10	当前请求返回的每页作业数量。
TotalCount	String	4998	当前可查询的历史作业的总数。

作业的状态码如下表

状态码	语义
starting	启动中，作业尚未执行。
running	作业在执行中。
error	作业中的程序执行失败，抛出了Exception。
dead	作业因为资源问题等问题失败。
killed	作业在执行中被取消。
success	作业执行成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ListSparkJob
&PageNumber=1
&PageSize=10
&VcName=MySparkCluster
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DataResult>
  <TotalCount>4998</TotalCount>
  <PageSize>2</PageSize>
  <PageNumber>1</PageNumber>
  <JobList>
    <ExecutorInstances>2</ExecutorInstances>
    <Status>success</Status>
    <VcName>MyCluster</VcName>
    <SparkUI>https://dlai-cn-hangzhou.aliyuncs.com/?token=xxx</SparkUI>
    <DriverResourceSpec>small</DriverResourceSpec>
    <CreateTime>2020-10-27 17:51:07</CreateTime>
    <JobName>localFile</JobName>
    <ExecutorResourceSpec>small</ExecutorResourceSpec>
    <SubmitTime>2020-10-27 17:50:43</SubmitTime>
    <CreateTimeValue>1603792267000</CreateTimeValue>
    <UpdateTimeValue>1603792302000</UpdateTimeValue>
    <SubmitTimeValue>1603792243000</SubmitTimeValue>
    <UpdateTime>2020-10-27 17:51:42</UpdateTime>
    <Detail>{"args":["100"],"name"f}</Detail>
    <JobId>j202010271750hangzhof742a4330000950</JobId>
  </JobList>
  <JobList>
    <ExecutorInstances>2</ExecutorInstances>
    <Status>error</Status>
    <VcName>MyCluster</VcName>
    <SparkUI>https://dlai-cn-hangzhou.aliyuncs.com/?token=xxx</SparkUI>
    <DriverResourceSpec>small</DriverResourceSpec>
    <CreateTime>2020-10-27 17:51:05</CreateTime>
    <JobName>localFile</JobName>
    <ExecutorResourceSpec>small</ExecutorResourceSpec>
    <SubmitTime>2020-10-27 17:50:39</SubmitTime>
    <CreateTimeValue>1603792265000</CreateTimeValue>
    <UpdateTimeValue>1603792300000</UpdateTimeValue>
    <SubmitTimeValue>1603792239000</SubmitTimeValue>
    <UpdateTime>2020-10-27 17:51:40</UpdateTime>
    <Detail>{"args":["100"],"name"f}</Detail>
    <JobId>j202010271750hangzhof742a4330000949</JobId>
  </JobList>
</DataResult>
<RequestId>E18B0634-F003-486A-90B9-4AA6932DA3AA</RequestId>
```

JSON 格式

```
{
  "DataResult": {
    "TotalCount": 4998,
    "PageSize": 2,
    "PageNumber": 1,
    "JobList": [
      {
        "ExecutorInstances": 2,
        "Status": "success",
        "VcName": "MyCluster",
        "SparkUI": "https://dlaui-cn-hangzhou.aliyuncs.com/?token=xxx",
        "DriverResourceSpec": "small",
        "CreateTime": "2020-10-27 17:51:07",
        "JobName": "localFile",
        "ExecutorResourceSpec": "small",
        "SubmitTime": "2020-10-27 17:50:43",
        "CreateTimeValue": 1603792267000,
        "UpdateTimeValue": 1603792302000,
        "SubmitTimeValue": 1603792243000,
        "UpdateTime": "2020-10-27 17:51:42",
        "Detail": "{\"args\": [\"100\"], \"name\": \"f\"}",
        "JobId": "j202010271750hangzhouf742a4330000950"
      },
      {
        "ExecutorInstances": 2,
        "Status": "error",
        "VcName": "MyCluster",
        "SparkUI": "https://dlaui-cn-hangzhou.aliyuncs.com/?token=xxx",
        "DriverResourceSpec": "small",
        "CreateTime": "2020-10-27 17:51:05",
        "JobName": "localFile",
        "ExecutorResourceSpec": "small",
        "SubmitTime": "2020-10-27 17:50:39",
        "CreateTimeValue": 1603792265000,
        "UpdateTimeValue": 1603792300000,
        "SubmitTimeValue": 1603792239000,
        "UpdateTime": "2020-10-27 17:51:40",
        "Detail": "{\"args\": [\"100\"], \"name\": \"f\"}",
        "JobId": "j202010271750hangzhouf742a4330000949"
      }
    ]
  },
  "RequestId": "E18B0634-F003-486A-90B9-4AA6932DA3AA"
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.9. ListSparkJobAttempt

获取一个Job对应的所有JobAttempt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	ListSparkJobAttempt	系统规定参数。取值：ListSparkJobAttempt。
JobId	String	是	j202105272322hangzhou5d64f1560000128	Spark作业ID。您可以在作业管理页面查看作业ID。
PageNumber	Integer	是	1	需要查询的页码号，用于遍历全部数据，页码的编码从1开始。
PageSize	Integer	是	10	每页返回的作业的个数。
VcName	String	是	release-test	执行Spark作业的虚拟集群名称。
Condition	Json	否	{ "status": "success", "createTime": "2021-05-27 11:00:00", "timeZone": "Asia/Shanghai" }	JobAttempt过滤条件，JSON格式。

Condition目前支持下列两个参数

- status: starting、running、unknown、success、dead、killed
- createTime: 格式为 yyyy-mm-dd HH:mm:ss，代表筛选指定时间之后提交的作业，指定该参数的时候，同时需要指定timeZone
- timeZone: 例如 Asia/Shanghai

返回数据

名称	类型	示例值	描述
DataResult	Struct		返回结果。
JobAttemptList	Array of Data		JobAttempt列表
CreateTime	String	2021-05-27 23:22:08	JobAttempt创建时间

名称	类型	示例值	描述
CreateTimeValue	String	1622128928000	JobAttempt创建时间戳
Detail	String	{"xxx":"xxx"}	作业配置
DurationTime	String	00:01:12	JobAttempt持续时间
DurationTimeValue	String	72000	持续时间戳
JobAttemptId	String	j202105272322hangzhou5d64f1560000128-0001	作业尝试ID
JobId	String	j202105272322hangzhou5d64f1560000128	作业ID
JobName	String	xxx	作业名称
SparkUI	String	xxx	SparkUI地址
Status	String	success	JobAttempt状态
TerminatedTime	String	2021-05-27 23:23:20	终止时间
TerminatedTimeValue	String	1622129000000	终止时间戳
VcName	String	release-test	虚拟集群名称
PageNumber	String	1	页数
PageSize	String	10	页大小
TotalCount	String	1	JobAttempt总数
RequestId	String	4A839055-70C7-423D-A6F2-E714EA4502DB	请求的唯一ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ListSparkJobAttempt
&JobId=j202105272322hangzhou5d64f1560000128
&PageNumber=1
&PageSize=10
&VcName=release-test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DataResult>
  <TotalCount>1</TotalCount>
  <PageSize>10</PageSize>
  <PageNumber>1</PageNumber>
  <JobAttemptList>
    <VcName>release-test</VcName>
    <Status>success</Status>
    <JobAttemptId>j202105272322hangzhou5d64f1560000128-0001</JobAttemptId>
    <TerminatedTime>2021-05-27 23:23:20</TerminatedTime>
    <SparkUI>xxx</SparkUI>
    <DurationTimeValue>72000</DurationTimeValue>
    <CreateTime>2021-05-27 23:22:08</CreateTime>
    <JobName>xxx</JobName>
    <TerminatedTimeValue>1622129000000</TerminatedTimeValue>
    <JobId>j202105272322hangzhou5d64f1560000128</JobId>
    <Detail>{"xxx":"xxx"}</Detail>
    <CreateTimeValue>1622128928000</CreateTimeValue>
    <DurationTime>00:01:12</DurationTime>
  </JobAttemptList>
</DataResult>
<RequestId>4A839055-70C7-423D-A6F2-E714EA4502DB</RequestId>
```

JSON 格式

```
{
  "DataResult": {
    "TotalCount": "1",
    "PageSize": "10",
    "PageNumber": "1",
    "JobAttemptList": [
      {
        "VcName": "release-test",
        "Status": "success",
        "JobAttemptId": "j202105272322hangzhou5d64f1560000128-0001",
        "TerminatedTime": "2021-05-27 23:23:20",
        "SparkUI": "xxx",
        "DurationTimeValue": "72000",
        "CreateTime": "2021-05-27 23:22:08",
        "JobName": "xxx",
        "TerminatedTimeValue": "1622129000000",
        "JobId": "j202105272322hangzhou5d64f1560000128",
        "Detail": "{\"xxx\":\"xxx\"}",
        "CreateTimeValue": "1622128928000",
        "DurationTime": "00:01:12"
      }
    ]
  },
  "RequestId": "4A839055-70C7-423D-A6F2-E714EA4502DB"
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.10. ListSparkStatements

获取Spark Session的所有代码块的执行状态。

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	ListSparkStatements	系统规定参数。取值：ListSparkStatements。
JobId	String	是	j202106071620hangzhou0000000001	Spark作业的ID。

返回数据

名称	类型	示例值	描述
RequestId	String	C2AB7692-B9EF-4A50-BCFF-0DC5B6F080E8	请求的唯一标识ID。
Statements	Array of Statements		代码块执行结果的列表。
Code	String	print(2 + 2)	提交执行的代码块。
CompletedTime	Long	1623033200754	执行完成的毫秒级时间戳。
Id	Integer	2	代码块的唯一ID。
Output	String	<pre>{"status\":\"ok\", \"execution_count\":1, \"data\": {\"text/plain\":\"4\"}}</pre>	代码块的执行结果，执行结果是一个JSON格式的字符串，其结构如下。 - status：执行状态。 - execution_count：当前执行到此代码块的哪一行。 - data：代码执行结果。
Progress	Float	1	当前执行进度，是一个浮点数，代表执行的百分比，'1'代表100%执行完成。
StartTime	Long	1623033200750	代码块开始执行的毫秒级时间戳。

名称	类型	示例值	描述
State	String	running	代码块的执行状态，可能的状态有： - waiting：等待执行。 - running：正在执行。 - available：执行完成。 - error：执行失败。 - cancelling：尝试终止。 - cancelled：已经终止执行。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ListSparkStatements
&JobId=j202106071620hangzhou00000000001
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>C2AB7692-B9EF-4A50-BCFF-0DC5B6F080E8</RequestId>
<Statements>
  <Progress>1</Progress>
  <CompletedTime>1623033200078</CompletedTime>
  <State>available</State>
  <Output>{"status":"ok","execution_count":0,"data":{"text/plain":"&lt;pyspark.sql.session.SparkSession object at 0x7f5e4a550390&gt;"},"}</Output>
  <StartTime>1623033200077</StartTime>
  <Id>0</Id>
  <Code>spark</Code>
</Statements>
<Statements>
  <Progress>1</Progress>
  <CompletedTime>1623033200754</CompletedTime>
  <State>available</State>
  <Output>{"status":"ok","execution_count":1,"data":{"text/plain":"4"},"}</Output>
  <StartTime>1623033200753</StartTime>
  <Id>1</Id>
  <Code>print(2 + 2)
</Code>
</Statements>
```

JSON 格式

```
{
  "RequestId": "C2AB7692-B9EF-4A50-BCFF-0DC5B6F080E8",
  "Statements": [
    {
      "Progress": 1,
      "CompletedTime": 1623033200078,
      "State": "available",
      "Output": "{\"status\": \"ok\", \"execution_count\": 0, \"data\": {\"text/plain\": \"<pyspark.sql.session.SparkSession object at 0x7f5e4a550390>\"}}",
      "StartTime": 1623033200077,
      "Id": 0,
      "Code": "spark"
    },
    {
      "Progress": 1,
      "CompletedTime": 1623033200754,
      "State": "available",
      "Output": "{\"status\": \"ok\", \"execution_count\": 1, \"data\": {\"text/plain\": \"4\"}}",
      "StartTime": 1623033200753,
      "Id": 1,
      "Code": "print(2 + 2)\\n"
    }
  ]
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.11. GetSparkStatement

获取指定代码块的执行状态。

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	GetSparkStateme nt	系统规定参数。取值： GetSparkStatement。
JobId	String	是	j202106071620ha ngzhou00000000 001	Spark作业的唯一id
StatementId	Integer	是	1	此Spark作业中代码块的唯一id

返回数据

名称	类型	示例值	描述
Data	Struct		描述了此段代码块的执行状态
Code	String	print(2+2)	提交的代码块
CompletedTime	Long	1623033200754	代码执行结束的毫秒级时间戳
Id	Integer	1	代码块在此Spark作业中的唯一递增id
Output	String	<pre>" {"status\":\"ok\", \" execution_count\":1 ,\"data\": {\"text/plain\":\"4\" }"} </pre>	代码块的执行结果, 执行结果是一个JSON格式的字符串, 其结构如下 - status 执行状态 - execution_count 当前执行到此代码块的哪一行 - data 代码执行结果
Process	Float	0.5	代码当前执行进度, 为浮点数, 1 代表100%执行完成, 0.5 代表执行了50%
StartTime	Long	1623033200750	代码开始执行的毫秒级时间戳
State	String	running	代码块的执行状态, 可能的状态有 - waiting 等待执行 - running 正在运行 - available 执行完成 - error 执行失败 - cancelling 在尝试终止 - cancelled 已经终止执行
RequestId	String	C5DB60E0-6EB3-493C-8956-DA0D3A2EE656	请求的唯一id

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetSparkStatement
&JobId=j202106071620hangzhou0000000001
&StatementId=1
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>C5DB60E0-6EB3-493C-8956-DA0D3A2EE656</RequestId>
<Data>
  <CompletedTime>1623033200754</CompletedTime>
  <State>available</State>
  <Output>{"status":"ok","execution_count":1,"data":{"text/plain":"4"}}</Output>
  <StartTime>1623033200753</StartTime>
  <Process>1</Process>
  <Id>1</Id>
  <Code>print(2 + 2)
</Code>
</Data>
```

JSON 格式

```
{
  "RequestId": "C5DB60E0-6EB3-493C-8956-DA0D3A2EE656",
  "Data": {
    "CompletedTime": 1623033200754,
    "State": "available",
    "Output": "{\"status\":\"ok\",\"execution_count\":1,\"data\":{\"text/plain\":\"4\"}}",
    "StartTime": 1623033200753,
    "Process": 1,
    "Id": 1,
    "Code": "print(2 + 2)\n"
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.12. ExecuteSparkStatement

调用ExecuteSparkStatement接口提交一段代码块给Spark作业执行。

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	ExecuteSparkStatement	系统规定参数。取值：ExecuteSparkStatement。

名称	类型	是否必选	示例值	描述
Code	String	是	print(2+2)\n	一段代码块，使用Linux系统的 '\n' 来代表切到新的一行。
JobId	String	是	j202106071620hangzhou****	Spark作业的ID。
Kind	String	是	sql	执行代码的语言是什么，取值： - scala - python - sql

返回数据

名称	类型	示例值	描述
Data	String	2	代码块提交成功后在此Spark作业中的唯一ID。
RequestId	String	0DC5B6F080E8-0DC5B6F080E8-0DC5B	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ExecuteSparkStatement
&Code=print(2+2)\n
&JobId=j202106071620hangzhou****
&Kind=sql
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>0DC5B6F080E8-0DC5B6F080E8-0DC5B</RequestId>
<Data>2</Data>
```

JSON 格式

```
{
  "RequestId": "0DC5B6F080E8-0DC5B6F080E8-0DC5B",
  "Data": 2
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.13. CancelSparkStatement

调用CancelSparkStatement接口终止一个代码块的执行。

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	CancelSparkStatement	系统规定参数。取值：CancelSparkStatement。
JobId	String	是	j202106071620hangzhou****	Spark作业的唯一ID。
StatementId	String	是	2026****	Spark作业中代码块的唯一ID。

返回数据

名称	类型	示例值	描述
Data	String	true	是否终止成功，取值说明： true：终止成功。 false：终止失败。
RequestId	String	341ADFA1-4A68-4707-9CA7-FA86523A5F14	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CancelSparkStatement
&JobId=j202106071620hangzhou****
&StatementId=2026****
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>341ADFA1-4A68-4707-9CA7-FA86523A5F14</RequestId>
<Data>true</Data>
```

JSON 格式

```
{
  "RequestId": "341ADFA1-4A68-4707-9CA7-FA86523A5F14",
  "Data": true
}
```

错误码

访问[错误中心](#)查看更多错误码。

此操作为异步操作, 当返回 `true` 时代表Spark作业收到了终止命令, 开始尝试终止代码块, 但最终能否成功依赖实际运行情况。

后续应当通过 `GetSparkStatement` 接口来查看最终是否终止成功。

6.14. GetSparkSessionState

调用GetSparkSessionState接口获取可交互Spark作业的状态。

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	GetSparkSessionState	系统规定参数。取值：GetSparkSessionState。
JobId	String	是	j202104171040zjk b4cc4****	可交互Spark作业ID。

返回数据

名称	类型	示例值	描述
RequestId	String	341ADFA1-4A68-4707-9CA7-FA86523A5F14	请求ID。
State	String	starting	当前Spark Session的状态, 取值说明： starting：Session在启动中。 idle：Session在空闲装填。 busy：Session在执行一段代码。 shutting_down：Session在关闭中。 error：Session发生内部错误，此时DLA会开始清理这个Session。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetSparkSessionState
&JobId=j202104171040zjkb4cc4****
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>341ADFA1-4A68-4707-9CA7-FA86523A5F14</RequestId>
<State>starting</State>
```

JSON 格式

```
{
  "RequestId": "341ADFA1-4A68-4707-9CA7-FA86523A5F14",
  "State": "starting"
}
```

错误码

访问[错误中心](#)查看更多错误码。

7.服务管理

7.1. CreateInstance

调用CreateInstance接口创建实例。

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	CreateInstance	系统规定参数。取值： CreateInstance 。
ChargeType	String	是	POSTPAY	固定为 POSTPAY 。暂只支持创建后付费实例。
Component	String	是	" {\"operable\": \"false\", \"max_cu\": \"8\", \"instance_name\": \"api-test-9\", \"min_cu\": \"0\", \"engine\": \"spark\", \"region\": \"cn-hangzhou\", \"app_name\": \"xx\", \"app_version_name\": \"xx\"}"	component模块的JSON序列化。
InstanceType	String	是	CU_PRESTO	实例类型，取值说明： CU_SPARK：云原生Spark引擎。 CU_PRESTO：兼容开源Presto语法。
RegionId	String	是	cn-beijing	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	600	错误码。

名称	类型	示例值	描述
ErrorInfo	String	An internal error occurred	错误信息。
RequestId	String	312F48C7-AFA0-46AA-B8F6-1B16B8808045	请求ID。
Result	String	true	返回的创建实例的结果。
Success	Boolean	true	请求是否调用成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CreateInstance
&ChargeType=POSTPAY
&Component="{\"operable\": \"false\", \"max_cu\": \"8\", \"instance_name\": \"api-test-9\", \"min_cu\": \"0\", \"engine\": \"spark\", \"region\": \"cn-hangzhou\", \"app_name\": \"xx\", \"app_version_name\": \"xx\"}"
&InstanceType=CU_PRESTO
&RegionId=cn-beijing
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>312F48C7-AFA0-46AA-B8F6-1B16B8808045</RequestId>
<ErrorInfo>An internal error occurred</ErrorInfo>
<ErrorCode>600</ErrorCode>
<Success>true</Success>
<Result>true</Result>
```

JSON 格式

```
{
  "RequestId": "312F48C7-AFA0-46AA-B8F6-1B16B8808045",
  "ErrorInfo": "An internal error occurred",
  "ErrorCode": 600,
  "Success": true,
  "Result": true
}
```

错误码

HttpCode	错误码	错误信息	描述
600	Openanalytics.Internal.Exception	An internal error occurred.	内部错误
401	Openanalytics.Unauthorized.Exception	You do not have permission to perform the operation.	您没有权限操作
404	Openanalytics.NotFound.Exception	The specified instance does not exist.	实例不存在

访问[错误中心](#)查看更多错误码。

7.2. ReleaseInstance

调用ReleaseInstance释放实例。

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	ReleaseInstance	系统规定参数。取值：ReleaseInstance。
InstanceId	String	是	dla-3qntmdtbqklsbsmceu****	实例ID。
RegionId	String	是	cn-beijing	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	600	错误码。
ErrorInfo	String	An internal error occurred	错误信息。
RequestId	String	312F48C7-AFA0-46AA-B8F6-1B16B8808045	请求ID。
Result	String	true	返回的释放实例的结果。

名称	类型	示例值	描述
Success	Boolean	true	请求是否调用成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ReleaseInstance
&InstanceId=dla-3qntmdtbqklsbsmceu***
&RegionId=cn-beijing
&<公共请求参数>
```

正常返回示例

XML 格式

```
<requestId>312F48C7-AFA0-46AA-B8F6-1B16B8808045</requestId>
<result>true</result>
<success>true</success>
```

JSON 格式

```
{"requestId":"312F48C7-AFA0-46AA-B8F6-1B16B8808045","result":"true","success":true}
```

错误码

HttpCode	错误码	错误信息	描述
600	Openanalytics.Internal.Exception	An internal error occurred.	内部错误
404	Openanalytics.NotFound.Exception	The specified instance does not exist.	实例不存在
401	Openanalytics.Unauthorized.Exception	You do not have permission to perform the operation.	您没有权限操作

访问[错误中心](#)查看更多错误码。

7.3. ValidateVirtualClusterName

调用ValidateVirtualClusterName校验虚拟集群的名字是否合法。

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	ValidateVirtualClusterName	系统规定参数。取值：ValidateVirtualClusterName。
VcName	String	否	abc123	需要校验的虚拟集群的名称。

返回数据

名称	类型	示例值	描述
Data	Struct		校验返回的数据。
Legal	String	true	校验返回值，是否合法。取值： true：合法 false：不合法
Message	String	null	如果虚拟集群名称不合法，上报的错误信息。
RequestId	String	C039FD4D-2F3C-4556-AF09-864D3A6485B2	请求ID。

示例

请求示例

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

正常返回示例

XML 格式

```
<RequestId>C039FD4D-2F3C-4556-AF09-864D3A6485B2</RequestId>
<Data>
  <Message>null</Message>
  <Legal>true</Legal>
</Data>
```

JSON 格式

```
{"RequestId":"C039FD4D-2F3C-4556-AF09-864D3A6485B2","Data":{"Message":"null","Legal":"true"}}
```

错误码

HttpCode	错误码	错误信息	描述
404	EmptyResult	The result is empty.	空结果。

访问[错误中心](#)查看更多错误码。

8.元数据

8.1. 权限管理

8.1.1. GrantPrivileges

调用Grant Privileges对用户进行授权。

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	GrantPrivileges	系统规定参数。取值：GrantPrivileges。
PrivilegeBag	Object	是		权限信息。
HiveObjectPrivilege	Array	是		权限信息的数组。
HiveObjectRef	Object	是		表名授权的粒度。
DbName	String	否	db001	数据库名称，如果需要给某个用户授予某张数据库或表的权限，需要指定这个字段。
HiveObjectType	String	是	GLOBAL	授权的粒度，目前仅支持GLOBAL、DATABASE以及TABLE，分别代表全局权限、库权限以及表权限。
TableName	String	否	tbl001	表的名称，如果需要给用户授予某张表的权限，需要指定这个字段。
PrincipalName	String	是	user001	被授权的用户。
PrincipalType	String	否	USER	只支持填写USER。
PrivilegeGrantInfo	Object	是		授权信息。

名称	类型	是否必选	示例值	描述
GrantOption	Boolean	否	false	被授权的用户可以把对应的权限授予给其他用户。
Grantor	String	是	user002	授权的用户的DLA userName。
PrincipalType	String	否	USER	固定取值为USER。
Privilege	String	是	CREATE	授予的具体权限。

返回数据

名称	类型	示例值	描述
Code	String	NO_PRIVILEGE	如果失败，则返回失败的代码。
Data	Boolean	false	授权是否成功。
Message	String	Access denied for user 'user002' to grant 'CREATE,SHOW' to user001	如果请求失败，则返回失败的详细信息。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	本次请求的RequestId。
Success	Boolean	false	本次请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GrantPrivileges
&PrivilegeBag={"HiveObjectPrivilege":[{"HiveObjectRef":{"DbName":"db001","HiveObjectType":"GLOBAL","TableName":"tbl001"},"PrincipalName":"user001","PrincipalType":"USER","PrivilegeGrantInfo":{"GrantOption":false,"Grantor":"user002","PrincipalType":"USER","Privilege":"CREATE"}}]}
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<Message>Access denied for user 'user002' to grant 'CREATE,SHOW' to user001</Message>
<RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
<Data>>false</Data>
<Code>NO_PRIVILEGE</Code>
<Success>>false</Success>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Message" : "Access denied for user 'user002' to grant 'CREATE,SHOW' to user001",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Data" : "false",
  "Code" : "NO_PRIVILEGE",
  "Success" : "false"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.1.2. RevokePrivileges

移除某个用户的某个权限

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
PrivilegeBag	Object	是		权限信息
HiveObjectPrivilege	Array	是		权限信息的数组
HiveObjectRef	Object	是		表明授权的粒度
DbName	String	是	db001	数据库名称，如果需要给某个用户授予某张数据库或表的权限，需要指定这个字段

名称	类型	是否必选	示例值	描述
HiveObjectType	String	是	GLOBAL	授权的粒度，目前仅支持 GLOBAL、DATABASE 以及 TABLE，分别代表全局权限、库权限以及表权限
TableName	String	是	tbl001	表的名称，如果需要给用户授予某张表的权限，需要指定这个字段
PrincipalName	String	是	user001	被授权的用户
PrincipalType	String	是	USER	只支持填写 USER
PrivilegeGrantInfo	Object	是		权限信息
GrantOption	Boolean	是	false	被授权的用户是够可以把对应的权限授予其他用户
Grantor	String	是	user002	授权的用户 DLA userName
PrincipalType	String	是	USER	只支持填写 USER
Privilege	String	是	CREATE	移除的具体权限

返回数据

名称	类型	示例值	描述
Code	String	NO_PRIVILEGE	如果失败。则返回失败的代码
Data	Boolean	false	是否移除权限成功
Message	String	Access denied for user 'user002' to revoke 'CREATE' to user001	如果请求失败，则返回失败的详细信息
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	本次请求的 RequestId

名称	类型	示例值	描述
Success	Boolean	false	本次请求是否成功

示例

请求示例

```
http(s)://[Endpoint]/?PrivilegeBag={"HiveObjectPrivilege":[{"PARAM_KEY":{"PrivilegeGrantInfo":{"GrantOption":"false","Grantor":"user002","Privilege":"CREATE","PrincipalType":"USER"},"HiveObjectRef":{"TableName":"tbl001","DbName":"db001","HiveObjectType":"GLOBAL"},"PrincipalName":"user001","PrincipalType":"USER"}}]}
```

&公共请求参数

正常返回示例

XML 格式

```
<Message>Access denied for user 'user002' to revoke 'CREATE' to user001</Message>
<RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
<Data>>false</Data>
<Code>NO_PRIVILEGE</Code>
<Success>>false</Success>
```

JSON 格式

```
{
  "Message": "Access denied for user 'user002' to revoke 'CREATE' to user001",
  "RequestId": "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Data": "false",
  "Code": "NO_PRIVILEGE",
  "Success": "false"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.2. 数据库管理

8.2.1. CreateDatabase

调用CreateDatabase创建数据库。

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	CreateDatabase	系统规定参数。取值：CreateDatabase。
Name	String	是	db001	数据库的名称。
Description	String	是	this is db description	数据库的描述。
LocationUri	String	是	oss://data/xxx	数据库的OSS地址。
Parameters	Map	否		数据库的属性。
	String	否	key001=value001	数据库的属性。

返回数据

名称	类型	示例值	描述
Code	String	ALREADY_EXISTS	标记错误的Code。
Data	Long	6666	表示创建成功时数据库的id。
Message	String	Database db001 already exists	如果有异常，返回异常的信息。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	本次请求的RequestId。
Success	Boolean	false	标记本次请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CreateDatabase
&Name=db001
&Description=this is db description
&LocationUri=oss://data/xxx
&Parameters={"key":"key001=value001"}
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<Message>Database db001 already exists</Message>
<RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
<Data>6666</Data>
<Code>ALREADY_EXISTS</Code>
<Success>false</Success>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Message" : "Database db001 already exists",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Data" : "6666",
  "Code" : "ALREADY_EXISTS",
  "Success" : "false"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.2.2. GetDatabase

调用GetDatabase获取数据库信息。

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	GetDatabase	系统规定参数。取值：GetDatabase。
Name	String	是	db001	需要获取的数据库名称。

返回数据

名称	类型	示例值	描述
Code	String	NO_SUCH_OBJECT	如果请求出错，则返回错误代码。

名称	类型	示例值	描述
DatabaseModel	Object		返回的数据库对象。
Description	String	this is db description	数据库描述信息。
LocationUri	String	oss://data/xxx	数据库用于存数据的OSS地址。
Name	String	db001	数据库名称。
OwnerName	String	user001	数据库是由谁创建的。
Parameters	Map		数据库属性。
	String	key001=value001	数据库属性。
TenantId	String	123455	阿里云的UID。
Message	String	The database db001 does not exist	如果请求出错，则返回错误详细信息。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	本次请求的RequestId。
Success	Boolean	false	本次请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetDatabase
&Name=db001
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<DatabaseModel>
  <LocationUri>oss://data/xxx</LocationUri>
  <OwnerName>user001</OwnerName>
  <TenantId>123455</TenantId>
  <Description>this is db description</Description>
  <Parameters>
    <key>key001=value001</key>
  </Parameters>
  <Name>db001</Name>
</DatabaseModel>
<Message>The database db001 does not exist</Message>
<RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
<Code>NO_SUCH_OBJECT</Code>
<Success>false</Success>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "DatabaseModel" : {
    "LocationUri" : "oss://data/xxx",
    "OwnerName" : "user001",
    "TenantId" : "123455",
    "Description" : "this is db description",
    "Parameters" : {
      "key" : "key001=value001"
    },
    "Name" : "db001"
  },
  "Message" : "The database db001 does not exist",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Code" : "NO_SUCH_OBJECT",
  "Success" : "false"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.2.3. GetAllDatabases

调用GetAllDatabases获取当前用户所有有权限的数据库。

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	GetAllDatabases	系统规定参数。取值： GetAllDatabases 。

返回数据

名称	类型	示例值	描述
Code	String	ok	如果有错误，则返回错误的类型。此API不会有异常。
Data	Array of String	db001	有权限的数据库名称列表。
Message	String	ok	如果有异常，则返回异常详细信息。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	本次请求的RequestId。
Success	Boolean	true	本次请求是否成功。

示例

请求示例

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

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<GetAllDatabasesResponse>
  <Code>ok</Code>
  <Data>db001</Data>
  <Message>ok</Message>
  <RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
  <Success>>true</Success>
</GetAllDatabasesResponse>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : "ok",
  "Data" : [ "db001" ],
  "Message" : "ok",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Success" : true
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.2.4. AlterDatabase

调用AlterDatabase接口修改数据库的信息。

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	AlterDatabase	系统规定参数。取值： AlterDatabase 。
Name	String	是	newDB001	新数据库的名称。
Description	String	是	This is database desc	数据库的描述。
LocationUri	String	是	oss://xxxx	数据库的存储路径。
OldDbName	String	是	oldDB001	旧数据库的名称。
Parameters	Map	否	key001=value=002	数据库的参数。
	String	否	key001=value=002	数据库的参数。

返回数据

名称	类型	示例值	描述
Code	String	NO_SUCH_OBJECT	标记成功错误的CODE。
Data	String	null	返回的数据。
Message	String	No such database: bd001	如果有错误，则返回错误的信息。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	请求ID。
Success	Boolean	true	修改数据库是否成功。

示例

请求示例

```
http(s)://openanalytics-open.aliyuncs.com/?Action=AlterDatabase
&Name=newDB001
&Description=This is database desc
&LocationUri=oss://xxxx
&OldDbName=oldDB001
&Parameters={"key":"key001=value=002"}
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<AlterDatabaseResponse>
  <Code>NO_SUCH_OBJECT</Code>
  <Data>null</Data>
  <Message>No such database: bd001</Message>
  <RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
  <Success>true</Success>
</AlterDatabaseResponse>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : "NO_SUCH_OBJECT",
  "Data" : "null",
  "Message" : "No such database: bd001",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Success" : true
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.2.5. DropDatabase

调用DropDatabase删除数据库。

 **说明** 此API接口无法删除数据库里面表的数据。

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	DropDatabase	系统规定参数。取值：DropDatabase。
Name	String	是	db001	需要删除的数据库名称。
Cascade	Boolean	是	false	取值： false：被删除的数据库里面有其他表的时候则删除失败。 true：被删除的数据库里面有其他表的时候，这些表将会被先删除，再删除对应的数据库。

返回数据

名称	类型	示例值	描述
Code	String	NO_SUCH_OBJECT	标记错误的Code。

名称	类型	示例值	描述
Data	String	null	无信息。
Message	String	No such database: bd001	如果删除数据库异常，则返回异常信息。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	本次请求的RequestId。
Success	Boolean	false	本次请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DropDatabase
&Name=db001
&Cascade=false
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<Message>No such database: bd001</Message>
<RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
<Data>null</Data>
<Code>NO_SUCH_OBJECT</Code>
<Success>false</Success>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Message" : "No such database: bd001",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Data" : "null",
  "Code" : "NO_SUCH_OBJECT",
  "Success" : "false"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.3. 分区管理

8.3.1. AddPartitions

调用AddPartitions在元信息系统中添加分区信息。

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	AddPartitions	系统规定参数。取值：AddPartitions。
Partition	Array	是		只发布国内站
DbName	String	是	db001	数据库名称。
Parameters	Map	否		分区参数。
	String	否	key001=value002	分区参数。
StorageDescriptor	Object	是		分区的SD信息。
Col	Array	是		表的列信息。
Comment	String	否	this is comment	列注释。
Name	String	否	col1	列名。
Type	String	否	string	列类型。
InputFormat	String	是	org.apache.hadoop.hive ql.io.parquet.MapredParquetInputFormat	表的InputFormat。
Location	String	否	oss://data/xxxx	表的路径。

名称	类型	是否必选	示例值	描述
OutputFormat	String	是	org.apache.hadoop.hive ql.io.parquet.MapredParquetOutputFormat	表的OutputFormat。
Parameters	Map	否		SD的属性。
	String	否	key001=value002	SD的属性。
SerdeInfo	Object	是		表的SerDe信息。
Name	String	否	serDeName	Serde的名称。
Parameters	Map	否		Serde的属性。
	String	否	key001=value002	Serde的属性。
SerializationLib	String	是	org.apache.hadoop.hive ql.io.parquet.serde.ParquetHiveSerDe	处理serDe的类库。
TableName	String	是	tbl001	表的名称。
Values	Array of String	是	p001	分区的值。

返回数据

名称	类型	示例值	描述
Code	String	INVALID_OPERATION	标记成功错误的CODE。
Data	Long	2	成功添加的分区数。
Message	String	Too many partitions, exceed max partition number	如果有错误，则返回错误的信息。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	请求RequestId。

名称	类型	示例值	描述
Success	Boolean	true	修改数据库是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=AddPartitions
&Partition=[{"DbName":"db001","Parameters":{"key":"key001=value002"},"StorageDescriptor":{"
Col":[{"Comment":"this is comment","Name":"col1","Type":"string"}],"InputFormat":"org.apach
e.hadoop.hive ql.io.parquet.MapredParquetInputFormat","Location":"oss://data/xxxx","OutputF
ormat":"org.apache.hadoop.hive ql.io.parquet.MapredParquetOutputFormat","Parameters":{"key"
:"key001=value002"},"SerDeInfo":{"Name":"serDeName","Parameters":{"key":"key001=value002"},
"SerializationLib":"org.apache.hadoop.hive ql.io.parquet.serde.ParquetHiveSerDe"}],"TableNa
me":"tbl001","Values":["p001"]}]]
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<Message>Too many partitions, exceed max partition number</Message>
<RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
<Data>2</Data>
<Code>INVALID_OPERATION</Code>
<Success>>true</Success>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Message" : "Too many partitions, exceed max partition number",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Data" : "2",
  "Code" : "INVALID_OPERATION",
  "Success" : "true"
}
```

错误码

HttpCode	错误码	错误信息	描述
430	ServerError.Inner	An internal error occurred.	系统内部错误

访问[错误中心](#)查看更多错误码。

8.3.2. GetPartition

调用GetPartition获取分区信息。

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	GetPartition	系统规定参数。取值：GetPartition。
DbName	String	是	db001	分区所在数据库名称。
TableName	String	是	tbl001	分区所在表名。
Values	Array of String	是	p001	获取的分区值。

返回数据

名称	类型	示例值	描述
Code	String	NO_SUCH_OBJECT	如果失败，则返回失败的代码。
Message	String	No such partition with values: [p001, not_exists] in table: db001.tbl001	如果请求失败，则返回失败的详细信息。
PartitionModel	Object		分区的详细信息。
CreateTime	Long	20180913	分区创建的时间。
DbName	String	db001	分区所在数据库名称。
Parameters	Map		分区属性。
	String	key001=value001	分区属性。

名称	类型	示例值	描述
StorageDescriptorModel	Object		分区SD信息。
Cols	Array of FieldSchemaModel		分区所在表的列信息。
Comment	String	this is comment	列注释。
Name	String	col1	列名称。
Type	String	string	列类型。
InputFormat	String	org.apache.hadoop.hive ql.io.parquet.M apredParquetInputFormat	表的InputFormat。
Location	String	oss://data/xxx	表的路径。
OutputFormat	String	org.apache.hadoop.hive ql.io.parquet.M apredParquetOutputFormat	表的OutputFormat。
Parameters	Map		表属性。
	String	key001=value001	表属性。
SerdeInfoModel	Object		表的SerDe信息。
Name	String	serDeName	Serde的名称。
Parameters	Map		Serde的属性。
	String	key001=value001	Serde的属性。
SerializationLib	String	org.apache.hadoop.hive ql.io.parquet.se rde.ParquetHiveSerDe	处理serde的类库。

名称	类型	示例值	描述
TableName	String	tbl001	分区所在的表名。
Values	Array of String	p001	分区的值。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	本次请求的RequestId。
Success	Boolean	false	本次请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetPartition
&DbName=db001
&TableName=tbl001
&Values=["p001"]
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<GetPartitionResponse>
  <Message>No such partition with values: [p001, not_exists] in table: db001.tbl001</Message>
  <RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
  <PartitionModel>
    <TableName>tbl001</TableName>
    <Parameters>
      <key>key001=value001</key>
    </Parameters>
    <CreateTime>123455</CreateTime>
    <Values>[ p001 ]</Values>
    <StorageDescriptorModel>
      <SerDeInfoModel>
        <SerializationLib>org.apache.hadoop.hive ql.io.parquet.serde.ParquetHiveSerDe</SerializationLib>
        <Parameters>
          <key>key001=value001</key>
        </Parameters>
        <Name>serDeName</Name>
      </SerDeInfoModel>
      <Cols>
        <FieldSchemaModel>
          <Comment>this is comment</Comment>
          <Type>string</Type>
          <Name>coll</Name>
        </FieldSchemaModel>
      </Cols>
      <Parameters>
        <key>key001=value001</key>
      </Parameters>
      <InputFormat>org.apache.hadoop.hive ql.io.parquet.MapredParquetInputFormat</InputFormat>
      <OutputFormat>org.apache.hadoop.hive ql.io.parquet.MapredParquetOutputFormat</OutputFormat>
      <Location>oss://data/xxx</Location>
    </StorageDescriptorModel>
    <DbName>db001</DbName>
  </PartitionModel>
  <Code>NO_SUCH_OBJECT</Code>
  <Success>false</Success>
</GetPartitionResponse>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Message" : "No such partition with values: [p001, not_exists] in table: db001.tbl001",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "PartitionModel" : {
    "TableName" : "tbl001",
    "Parameters" : {
      "key" : "key001=value001"
    },
    "CreateTime" : "123455",
    "Values" : "[ p001 ]",
    "StorageDescriptorModel" : {
      "SerDeInfoModel" : {
        "SerializationLib" : "org.apache.hadoop.hive ql.io.parquet.serde.ParquetHiveSerDe",
        "Parameters" : {
          "key" : "key001=value001"
        },
        "Name" : "serDeName"
      },
      "Cols" : [ {
        "FieldSchemaModel" : {
          "Comment" : "this is comment",
          "Type" : "string",
          "Name" : "col1"
        }
      } ],
      "Parameters" : {
        "key" : "key001=value001"
      },
      "InputFormat" : "org.apache.hadoop.hive ql.io.parquet.MapredParquetInputFormat",
      "OutputFormat" : "org.apache.hadoop.hive ql.io.parquet.MapredParquetOutputFormat",
      "Location" : "oss://data/xxx"
    },
    "DbName" : "db001"
  },
  "Code" : "NO_SUCH_OBJECT",
  "Success" : "false"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.3.3. GetPartitions

调用GetPartitions获取表的所有分区信息。

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	GetPartitions	系统规定参数。取值：GetPartitions。
DbName	String	是	db001	数据库名称。
TableName	String	是	tbl001	表的名称。
MaxParts	Integer	否	100	每次最多返回多少个分区。默认为 -1，代表返回所有分区。

返回数据

名称	类型	示例值	描述
Code	String	NO_SUCH_OBJECT	如果失败，则返回失败的代码。
Message	String	No such table: db001.tbl001	如果请求失败，返回详细的失败信息。
Partitions	Array of PartitionModel		分区列表。
CreateTime	Long	20180913	分区创建时间。
DbName	String	db001	分区所在数据库名称。
Parameters	Map		分区属性。
	String	key001=value001	分区属性。
StorageDescriptorModel	Object		分区SD信息。
Cols	Array of FieldSchemaModel		分区所在表的列信息。
Comment	String	this is comment	列注释。

名称	类型	示例值	描述
Name	String	col1	列名称。
Type	String	string	列类型。
InputFormat	String	org.apache.hadoop.hive ql.io.parquet.M apredParquetInputF ormat	表的InputFormat。
Location	String	oss://data/xxx	表的路径。
OutputFormat	String	org.apache.hadoop.hive ql.io.parquet.M apredParquetOutpu tFormat	表的OutputFormat。
Parameters	Map		SD属性。
	String	key001=value001	SD属性。
SerDeInfoModel	Object		表的SerDe信息。
Name	String	serDeName	SerDe的名称。
Parameters	Map		SerDe的属性。
	String	key001=value001	SerDe的属性。
SerializationLib	String	org.apache.hadoop.hive ql.io.parquet.se rde.ParquetHiveSerD e	处理serDe的类库。
TableName	String	tbl001	分区所在的表名。
Values	Array of String	p001	分区的值。
RequestId	String	9BEAC206-0795- 4DE3-B1FD- 964BEF432B23	本次请求的RequestId。

名称	类型	示例值	描述
Success	Boolean	false	本次请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetPartitions
&DbName=db001
&TableName=tbl001
&MaxParts=100
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<GetPartitionsResponse>
  <Code>NO_SUCH_OBJECT</Code>
  <Message>No such table: db001.tbl001</Message>
  <Partitions>
    <CreateTime>20180913</CreateTime>
    <DbName>db001</DbName>
    <Parameters>
      <key>key001=value001</key>
    </Parameters>
    <StorageDescriptorModel>
      <Cols>
        <Comment>this is comment</Comment>
        <Name>col1</Name>
        <Type>string</Type>
      </Cols>
      <InputFormat>org.apache.hadoop.hive ql.io.parquet.MapredParquetInputFormat</InputFormat>
      <Location>oss://data/xxx</Location>
      <OutputFormat>org.apache.hadoop.hive ql.io.parquet.MapredParquetOutputFormat</OutputFormat>
      <Parameters>
        <key>key001=value001</key>
      </Parameters>
      <SerDeInfoModel>
        <Name>serDeName</Name>
        <Parameters>
          <key>key001=value001</key>
        </Parameters>
        <SerializationLib>org.apache.hadoop.hive ql.io.parquet.serde.ParquetHiveSerDe</SerializationLib>
      </SerDeInfoModel>
    </StorageDescriptorModel>
    <TableName>tbl001</TableName>
    <Values>p001</Values>
  </Partitions>
  <RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
  <Success>false</Success>
</GetPartitionsResponse>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : "NO_SUCH_OBJECT",
  "Message" : "No such table: db001.tbl001",
  "Partitions" : [ {
    "CreateTime" : 20180913,
    "DbName" : "db001",
    "Parameters" : {
      "key" : "key001=value001"
    },
    "StorageDescriptorModel" : {
      "Cols" : [ {
        "Comment" : "this is comment",
        "Name" : "col1",
        "Type" : "string"
      } ],
      "InputFormat" : "org.apache.hadoop.hive ql.io.parquet.MapredParquetInputFormat",
      "Location" : "oss://data/xxx",
      "OutputFormat" : "org.apache.hadoop.hive ql.io.parquet.MapredParquetOutputFormat",
      "Parameters" : {
        "key" : "key001=value001"
      },
      "SerDeInfoModel" : {
        "Name" : "serDeName",
        "Parameters" : {
          "key" : "key001=value001"
        },
        "SerializationLib" : "org.apache.hadoop.hive ql.io.parquet.serde.ParquetHiveSerDe"
      }
    },
    "TableName" : "tbl001",
    "Values" : [ "p001" ]
  } ],
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Success" : false
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.3.4. DropPartition

调用DropPartition删除分区信息。

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	DropPartition	系统规定参数。取值：DropPartition。
DbName	String	是	db001	需要删除分区所在数据库名称。
TableName	String	是	tbl001	需要删除分区所在数据表的名称。
PartValues	Array of String	是	p002	需要删除的分区。

返回数据

名称	类型	示例值	描述
Code	String	NO_SUCH_OBJECT	如果请求出错则返回出错的代码。
Data	Boolean	false	删除分区是否成功。
Message	String	Table: tbl001 not exists!	如果删除分区出错，则返回错误信息。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	本次请求的RequestId。
Success	Boolean	false	本次请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DropPartition
&DbName=db001
&TableName=tbl001
&PartValues=["p002"]
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<Message>Table: tbl001 not exists!</Message>
<RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
<Data>>false</Data>
<Code>NO_SUCH_OBJECT</Code>
<Success>>false</Success>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Message" : "Table: tbl001 not exists!",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Data" : "false",
  "Code" : "NO_SUCH_OBJECT",
  "Success" : "false"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.4. 表管理

8.4.1. CreateTable

调用CreateTable创建OSS的表。

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	CreateTable	系统规定参数。取值：CreateTable。
DbName	String	是	db001	数据库的名称。
TableName	String	是	tbl001	需要创建的表名。
PartitionKeys	Array	否		分区字段信息，可选参数。

名称	类型	是否必选	示例值	描述
Comment	String	否	this is partition keys	分区注释。
Name	String	是	part_id001	分区的列名。
Type	String	是	string	分区列类型。
Parameters	Map	否		表属性。
	String	否	key001=value001	表属性。
StorageDescriptor	Object	是		表的存储描述信息。
Cols	Array	是		表的列信息。
Comment	String	否	this is column comment	表的列字段注释。
Name	String	是	col1	表的列名称。
Type	String	是	string	表的列类型。
InputFormat	String	是	org.apache.hadoop.hive ql.io.parquet.MapredParquetInputFormat	表的InputFormat。
Location	String	是	oss://data/warehouse/tblname	表数据存储的目录。
OutputFormat	String	是	org.apache.hadoop.hive ql.io.parquet.MapredParquetOutputFormat	表的OutputFormat信息。
Parameters	Map	否		SD属性。
	String	否	key001=value001	SD属性。
SerdeInfo	Object	是		表的序列化和反序列信息。

名称	类型	是否必选	示例值	描述
Name	String	否	serDename	表的序列化和反序列化的名称。
Parameters	Map	否		表的序列化和反序列化的属性。
	String	否	key001=value001	表的序列化和反序列化的属性。
SerializationLib	String	是	org.apache.hadoop.hive ql.io.parquet.serde.ParquetHiveSerDe	表的序列化和反序列化的类库。
ViewOriginalText	String	否	select count(*) from tbl001	视图的原始SQL。
ViewExpandedText	String	否	select count(*) from tbl001	视图的展开SQL。
TableType	String	是	VIEW	表的类型，比如VIEW、MANAGED_TABLE、EXTERNAL_TABLE、VIRTUAL_VIEW等。

返回数据

名称	类型	示例值	描述
Code	String	ALREADY_EXISTS	标记错误的Code。
Data	Long	1234	成功创建表时返回表的ID。
Message	String	Table db001.tbl001 already exists	如果创建表出错，返回错误的详细信息。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	本次请求的RequestId。
Success	Boolean	false	标记本次请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CreateTable
&DbName=db001
&TableName=tbl001
&PartitionKeys=[{"Comment":"this is partition keys","Name":"part_id001","Type":"string"}]
&Parameters={"key":"key001=value001"}
&StorageDescriptor={"Cols":[{"Comment":"this is column comment","Name":"col1","Type":"string"}],
"InputFormat":"org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat",
"Location":"oss://data/warehouse/tblname",
"OutputFormat":"org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat",
"Parameters":{"key":"key001=value001"},
"SerDeInfo":{"Name":"serDenam",
"Parameters":{"key":"key001=value001"},
"SerializationLib":"org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe"}}
&ViewOriginalText=select count(*) from tbl001
&ViewExpandedText=select count(*) from tbl001
&TableType=VIEW
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<Message>Table db001.tbl001 already exists</Message>
<RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
<Data>1234</Data>
<Code>ALREADY_EXISTS</Code>
<Success>>false</Success>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Message" : "Table db001.tbl001 already exists",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Data" : "1234",
  "Code" : "ALREADY_EXISTS",
  "Success" : "false"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.4.2. GetTable

调用GetTable获取给定表的详细信息。

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	GetTable	系统规定参数。取值：GetTable。
DbName	String	是	db001	数据库名称。
TableName	String	是	tbl001	表名称。

返回数据

名称	类型	示例值	描述
Code	String	NO_SUCH_OBJECT	如果失败，则返回失败的代码。
Message	String	No such table: db001.tbl001	如果请求失败，则返回失败的详细信息。
RequestId	String	9BEAC206-0795- 4DE3-B1FD- 964BEF432B23	本次请求的RequestId。
Success	Boolean	false	本次请求是否成功。
Table	Object		获取的表信息。
CreateTime	Long	20190813	表的创建时间。
DbName	String	db001	数据库名称。
LastAccessTime	Long	20190814	表的最后访问时间。
Owner	String	user001	创建表的DLA username。
Parameters	Map		表的属性。
	String	key001=value001	表的属性。
PartitionKeys	Array of PartitionKeys		表的分区信息。

名称	类型	示例值	描述
Comment	String	this is partition comment	分区注释。
Name	String	part001	分区字段名称。
Type	String	int	分区列类型。
StorageDescriptor	Object		表的SD信息。
Cols	Array of Cols		表的列信息。
Comment	String	this is column comment	列注释。
Name	String	col1	列名称。
Type	String	string	列类型。
InputFormat	String	org.apache.hadoop.hive ql.io.parquet.MapredParquetInputFormat	表的InputFormat。
Location	String	oss://data/xxxx	表的路径。
OutputFormat	String	org.apache.hadoop.hive ql.io.parquet.MapredParquetOutputFormat	表的OutputFormat。
Parameters	Map		SD属性。
SerdeInfo	Object		Serde信息。
Name	String	SerdeName	Serde名字。
Parameters	Map		Serde属性。
	String	key001=value001	Serde属性。

名称	类型	示例值	描述
SerializationLib	String	org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe	处理serDe的类库。
TableName	String	tbl001	表名称。
TableType	String	VIEW	表类型，支持VIEW、MANAGED_TABLE、EXTERNAL_TABLE、VIRTUAL_VIEW。
ViewExpandedText	String	select count(*) from tbl002	如果表是视图，则代表视图的扩展SQL。
ViewOriginalText	String	select count(*) from tbl002	如果表是视图，则代表视图的原始SQL。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetTable
&DbName=db001
&TableName=tbl001
&公共请求参数
```

正常返回示例

JSON 格式

```
HTTP/1.1 200 OK
```

错误码

访问[错误中心](#)查看更多错误码。

8.4.3. GetAllTables

调用GetAllTables获取数据库下面的所有有权限的表。

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
----	----	------	-----	----

名称	类型	是否必选	示例值	描述
Action	String	是	GetAllTables	系统规定参数。取值：GetAllTables。
DbName	String	是	db001	需要获取数据库的名称。

返回数据

名称	类型	示例值	描述
Code	String	NO_SUCH_OBJECT	如果请求有异常，则返回异常编码。
Data	Array of String	tbl001	如果请求成功，则返回db001下面的所有有权限的表。
Message	String	The database db001 does not exist	如果出现异常，则返回错误的信息。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	本次请求的RequestId。
Success	Boolean	false	本次请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetAllTables
&DbName=db001
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<Message>The database db001 does not exist</Message>
<RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
<Data>[ tbl001,tbl002 ]</Data>
<Code>NO_SUCH_OBJECT</Code>
<Success>>false</Success>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Message" : "The database db001 does not exist",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Data" : "[ tbl001,tbl002 ]",
  "Code" : "NO_SUCH_OBJECT",
  "Success" : "false"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.4.4. AlterTable

调用AlterTable修改表的数据库名、表名、表属性以及列信息。

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	AlterTable	系统规定参数。取值：AlterTable。
OldDbName	String	是	oldDb001	旧数据库的名称。
OldTableName	String	是	oldTbl001	旧表的名称。
NewDbName	String	是	newDb001	新数据库的名称，可以和旧数据库名称一样，代表不修改数据库名。
NewTableName	String	是	newTbl001	新表的名称，可以和旧表名称一样，代表不修改表名。
Parameters	Map	否		表属性。
	String	否	key001=value001	表属性。
Col	Array	是		列信息。

名称	类型	是否必选	示例值	描述
Comment	String	否	this is comment	列注释。
Name	String	是	new_col1	列名称。
Type	String	是	int	列类型。

返回数据

名称	类型	示例值	描述
Code	String	NO_SUCH_OBJECT	标记错误的CODE。
Data	Boolean	false	标记修改表是否成功。
Message	String	No such database: bd001	如果有错误，则返回错误的信息。
RequestId	String	9BEAC206-0795- 4DE3-B1FD- 964BEF432B23	本次请求的RequestId。
Success	Boolean	true	本次请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=AlterTable
&OldDbName=oldDb001
&OldTableName=oldTbl001
&NewDbName=newDb001
&NewTableName=newTbl001
&Parameters={"key":"key001=value001"}
&Col=[{"Comment":"this is comment","Name":"new_col1","Type":"int"}]
&公共请求参数
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<Message>No such database: bd001</Message>
<RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
<Data>false</Data>
<Code>NO_SUCH_OBJECT</Code>
<Success>true</Success>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Message" : "No such database: bd001",
  "RequestId" : "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Data" : "false",
  "Code" : "NO_SUCH_OBJECT",
  "Success" : "true"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.4.5. DropTable

将给定表删除

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
DbName	String	是	db001	需要删除表所在的数据库
TableName	String	是	tbl001	需要删除的表

返回数据

名称	类型	示例值	描述
Code	String	NO_SUCH_OBJECT	标记成功错误的 CODE

名称	类型	示例值	描述
Data	Boolean	false	标记是否删除成功
Message	String	No such table: db001.tbl001	如果删除表失败，返回失败的原因。
RequestId	String	9BEAC206-0795-4DE3-B1FD-964BEF432B23	请求的 RequestId
Success	Boolean	false	标记本次请求是否成功

示例

请求示例

```
http(s)://[Endpoint]/?TableName=tbl001
&DbName=db001
&公共请求参数
```

正常返回示例

XML 格式

```
<Message>No such table: db001.tbl001</Message>
<RequestId>9BEAC206-0795-4DE3-B1FD-964BEF432B23</RequestId>
<Data>>false</Data>
<Code>NO_SUCH_OBJECT</Code>
<Success>>false</Success>
```

JSON 格式

```
{
  "Message": "No such table: db001.tbl001",
  "RequestId": "9BEAC206-0795-4DE3-B1FD-964BEF432B23",
  "Data": "false",
  "Code": "NO_SUCH_OBJECT",
  "Success": "false"
}
```

错误码

访问[错误中心](#)查看更多错误码。