阿里云

专有云企业版

数据管理 开发指南

产品版本: v3.16.2

文档版本: 20220915

(一)阿里云

数据管理 开发指南·法律声明

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。

数据管理 开发指南·通用约定

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。	
□ 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	八)注意 权重设置为0,该服务器不会再接受新请求。
⑦ 说明	用于补充说明、最佳实践、窍门等 <i>,</i> 不是用户必须了解的内容。	② 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 listinstanceid 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}

开发指南·目录

目录

1.快速入门	09
1.1. 开发文档使用说明	09
1.2. API调用流程说明	10
2.准备工作	13
2.1. 登录API与工具控制台	13
2.2. 查看API信息	13
2.3. 获取AccessKey	14
2.4. 获取STS AccessKey	15
2.5. 获取DMS的Endpoint	19
2.6. 获取DMS的SDK包	19
3.Java SDK调用示例	21
3.1. 快速开始	21
3.2. 使用手册	22
3.2.1. 安装Java SDK	22
3.2.2. 设置身份验证凭证	23
3.2.3. 请求连接配置	23
3.2.4. 发起调用	24
3.2.5. 错误处理	25
4.Python SDK调用示例	27
4.1. 快速开始	27
4.2. 使用手册	27
4.2.1. 安装Python SDK	27
4.2.2. 设置身份验证凭证	28
4.2.3. 设置HTTPS请求	28
4.2.4. 发起调用	29
4.2.5. 错误处理	29

5.Go SDK调用示例	31
5.1. 快速开始	31
5.2. 使用手册	32
5.2.1. 安装Go SDK核心包	32
5.2.2. 设置身份验证凭证	32
5.2.3. 发起调用	33
5.2.4. 错误处理	33
6.API参考	35
6.1. API概览	35
6.2. 用户管理	37
6.2.1. RegisterUser	38
6.2.2. DisableUser	39
6.2.3. EnableUser	41
6.2.4. UpdateUser	42
6.2.5. DeleteUser	43
6.2.6. ListUsers	45
6.2.7. ListUserTenants	49
6.2.8. GetUserActiveTenant	51
6.2.9. GetUser	53
6.3. 实例管理	56
6.3.1. RegisterInstance	56
6.3.2. ListInstances	61
6.3.3. GetInstance	65
6.3.4. UpdateInstance	69
6.3.5. DeleteInstance	72
6.4. 操作日志	73
6.4.1. GetOpLog	7 3
6.5. 审批流程	83

	6.5.1. ListWorkFlowTemplates	83
	6.5.2. ListWorkFlowNodes	- 88
6.	.6. 库表元数据	90
	6.6.1. SearchDatabase	90
	6.6.2. SearchTable	95
	6.6.3. ListDatabases	100
	6.6.4. ListLogicDatabases	104
	6.6.5. ListTables	108
	6.6.6. ListLogicTables	112
	6.6.7. ListColumns	116
	6.6.8. ListIndexes	121
	6.6.9. GetDatabase	123
	6.6.10. GetLogicDatabase	127
	6.6.11. GetMetaTableDetailInfo	130
	6.6.12. GetTableDBTopology	134
	6.6.13. GetMetaTableColumn	139
	6.6.14. SyncDatabaseMeta	142
	6.6.15. SyncInstanceMeta	144
6.	.7. 工单操作	145
	6.7.1. 通用	145
	6.7.1.1. ApproveOrder	145
	6.7.1.2. CloseOrder	146
	6.7.1.3. GetApprovalDetail	148
	6.7.1.4. GetOrderBaseInfo	151
	6.7.1.5. ListOrders	154
	6.7.1.6. SubmitOrderApproval	159
	6.7.1.7. CreateOrder	160
	6.7.2. 数据变更	164

6.7.2.1. ExecuteDataCorrect	164
6.7.2.2. GetDataCorrectBackupFiles	166
6.7.2.3. GetDataCorrectOrderDetail	167
6.7.3. 结构设计	171
6.7.3.1. CreatePublishGroupTask	171
6.7.4. 数据导出	173
6.7.4.1. ExecuteDataExport	173
6.7.4.2. GetDataExportDownloadURL	175
6.7.4.3. GetDataExportOrderDetail	176
6.8. 权限管理	179
6.8.1. RevokeUserPermission	179
6.8.2. ListDatabaseUserPermssions	181
6.8.3. SetOwners	185
6.8.4. ListUserPermissions	187
6.8.5. GrantUserPermission	192
6.9. 安全管理	194
6.9.1. ListSensitiveColumns	194
6.9.2. ListSensitiveColumnsDetail	198
6.10. 数据查询	201
6.10.1. ExecuteScript	201
7.附录	204
7.1. ASAPI网关调用流程说明	204
7.2. 通过ASAPI网关调用专有云API	206
7.2.1. 获取ASAPI的Endpoint	206
7.2.2. 获取ASAPI的Endpoint	211
7.2.3. 获取ASAPI的SDK包	211
7.3. STS概要	212
7.3.1. 什么是STS	212

数据管理 开发指南·<mark>目录</mark>

7.3.2. AssumeRole ------ 213

数据管理 开发指南·快速入门

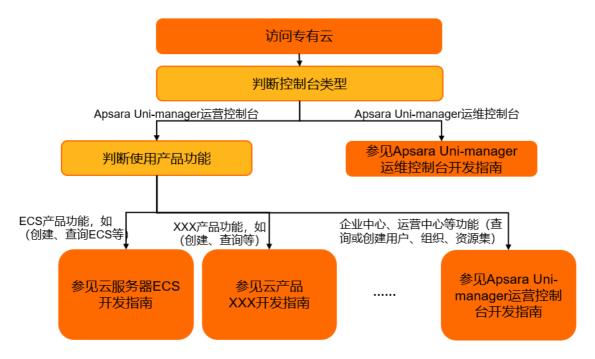
1.快速入门

1.1. 开发文档使用说明

本文为您介绍在使用专有云API进行开发工作时,如何查看所需要的产品文档。

使用流程

您可以按照以下流程图中步骤,判断自己所需要使用的开发文档。



前提条件

通过运维人员或运营管理员获取专有云的版本。

操作步骤

您可以通过阿里云专有云文档中心,获取需要的产品文档。

- 1. 通过浏览器打开专有云文档中心。
- 2. 根据专有云的版本单击企业版、敏捷版或一体机标签。

开发指南·快速入门 数据管理

3. 在下拉列表中,选择对应的专有云版本。



- 4. 在左侧导航栏中,选择目标产品名称。
- 5. 在单击目标文档区域的查看详情或下载PDF。

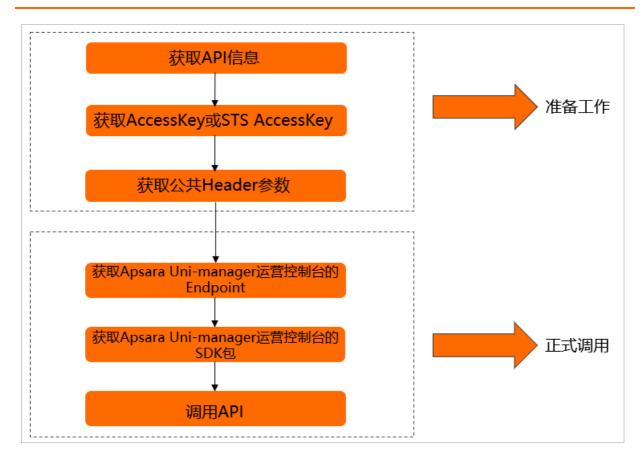
1.2. API调用流程说明

阿里云专有云支持通过POP网关或ASAPI网关调用专有云API,为了使您拥有与阿里云公有云一致的体验,自企业版V3.16.0开始,专有云API默认使用POP网关方式调用。本文介绍通过POP网关调用专有云API的流程。



历史版本使用ASAPI网关调用专有云API的方式仍然有效,详情请参见ASAPI<mark>网关调用流程说明、获取ASAPI的Endpoint、获取ASAPI的SDK包</mark>。

流程图

调用步骤

1. 获取AccessKey。

获取调用时所需要的AccessKey参数,请参见获取AccessKey或获取STS AccessKey。

? 说明

推荐使用个人AccessKey或STS AccessKey调用Apsara Uni-manager运营控制台及云产品接口。

2. 获取RegionID。

获取调用时所需要的RegionID参数,请参见获取公共Header参数。

3. 获取组织ID与资源集ID。

资源集ID需要通过API进行获取,获取方式请参见*Apsara Uni-manager运营控制台开发指南*中的*List Reso urceGroup章节*。

开发指南·快速入门 数据管理



4. 获取对应的API参考文档。

您需要查看目标云产品的开发指南,了解API的功能与使用方式,开发指南的使用流程及获取方法请参见开发文档使用说明。

- 5. 获取云产品Endpoint。
- 6. 获取云产品SDK并调用云产品API。

数据管理 开发指南·准备工作

2.准备工作

2.1. 登录API与工具控制台

您可以通过Apsara Uni-manager运营控制台登录API与工具控制台。

前提条件

- 登录Apsara Uni-manager运营控制台前,确认您已从部署人员处获取Apsara Uni-manager运营控制台的服务域名地址。
- 推荐使用Chrome浏览器

操作步骤

- 1. 在浏览器地址栏中,输入Apsara Uni-manager运营控制台的服务域名地址,按回车键。
- 2. 输入正确的用户名及密码。

请向运营管理员获取登录控制台的用户名和密码。

- ② 说明 首次登录Apsara Uni-manager运营控制台时,需要修改登录用户名的密码,请按照提示完成密码修改。为提高安全性,密码长度必须为10~32位,且至少包含以下两种类型:
 - 英文大写或小写字母(A~Z、a~z)
 - 阿拉伯数字(0~9)
 - 特殊符号(感叹号(!)、at(@)、井号(#)、美元符号(\$)、百分号(%)等)
- 3. 单击登录。
- 4. 如果账号已激活MFA多因素认证,请根据以下两种情况进行操作:
 - 管理员强制开启MFA后的首次登录:
 - a. 在绑定虚拟MFA设备页面中,按页面提示步骤绑定MFA设备。
 - b. 按照步骤2重新输入账号和密码,单击登录。
 - c. 输入6位MFA码后单击**认证**。
 - 您已开启并绑定MFA:

输入6位MFA码后单击**认证**。

- ② 说明 绑定并开启MFA的操作请参见Apsara Uni-manager运营控制台用户指南中的绑定并开启虚拟MFA设备章节。
- 5. 在页面顶部的菜单栏中,选择产品 > 其他 > API与工具。

2.2. 查看API信息

您需要获取产品名称、API名称和API版本,使用SDK时需要填写这些信息。

背景信息

API目录中的API仅供调用参考,具体API的使用方式需要参见云产品的开发指南。如果发生不一致情况,以开

开发指南·准备工作 数据管理

发指南支持情况为准。

操作步骤

- 1. 登录API与工具控制台。
- 2. 在左侧导航栏中选择API目录。
- 3. 在API目录页面,在下拉列表框中选择对应的产品,输入API名称的关键词进行搜索。
- 4. 查看并记录对应的产品名称、API名称和API版本。

2.3. 获取AccessKey

AccessKey支持RAM和STS两种授权模式,在发起调用时选择使用其中一种AccessKey即可。本文将为您介绍如何获取通过RAM授权的AccessKey。

背景信息

AccessKey通过使用AccessKey ID和AccessKey Secret对称加密,实现验证某个请求发送者的身份的功能。 其中AccessKey ID用于标示用户,AccessKey Secret是用户用于加密签名的字符串。

AccessKey的两种授权模式如下:

- 通过RAM授权给第三方请求者的AccessKey ID和AccessKey Secret。
- 通过STS授权给第三方请求者的AccessKey ID、AccessKey Secret和Security Token。获取方式请参见获取 STS AccessKey。

Apsara Uni-manager运营控制台提供了个人AccessKey和组织AccessKey(只有运营管理员和一级组织管理员可以获取组织AccessKey),推荐使用个人账号AccessKey调用Apsara Uni-manager运营控制台及云产品相关接口。如果使用个人账号AccessKey,需要在Header中添加以下限制性参数,否则可能提示权限不足:

● 警告 个人账号AccessKey是由Apsara Uni-manager运营控制台权限体系管控的受限AccessKey,组织AccessKey相对权限较大,需要管理员确认操作的安全性。

参数名称	描述
x-acs-regionid	地域,如:cn-hangzhou-*。
x-acs-organizationid	Apsara Uni-manager运营控制台中对应的组织ID。获取 方法可参见 <i>Apsara Uni-manager运营控制台开发指南</i> 中 的 <i>GetOrganizationList</i> 文档。
x-acs-resourcegroupid	Apsara Uni-manager运营控制台中对应的资源集ID。获取方法可参见 <i>Apsara Uni-manager运营控制台开发指南</i> 中的 <i>ListResourceGroup</i> 文档。
x-acs-instanceid	进行操作的实例ID,例如,ECS的实例ID可以从ECS列表中查询到。

获取个人账号AccessKey

获取个人账号AccessKey的方法如下:

- 1. 登录Apsara Uni-manager运营控制台。
- 2. 在系统界面右上角,单击当前登录用户头像,单击个人信息。

 数据管理 开发指南·准备工作

3. 在阿里云AccessKey区域,您可以查看个人账户的AccessKey信息。



② 说明 Accesskey ID和AccessKey Secret是您访问云资源时的密钥,具有该账号完整的权限,请您妥善保管。

获取组织AccessKey

获取组织AccessKey的方法如下:

- 1. 管理员登录Apsara Uni-manager运营控制台。
- 2. 在顶部菜单栏,单击企业。
- 3. 在左侧导航栏中,选择资源管理 > 组织管理。
- 4. 在组织结构中,单击目标一级组织名称。
- 5. 单击管理Accesskey。
- 6. 在弹出的对话框中,查看组织AccessKey信息。

2.4. 获取STS AccessKey

本文为您介绍如何利用STS方式,调用专有云的产品API。

背景信息

阿里云临时安全令牌STS(Security Token Service)是阿里云提供的一种临时访问权限管理服务。您可以选择使用STS方式,调用专有云的产品API。

STS AccessKey由AccessKey ID、AccessKey Secret及SecurityToken三元组组成,使用STS方式调用云产品API时,三元组缺一不可。更多STS内容请参见什么是STS。

操作步骤

- 1. 获取用户角色的 RAM Role 。
 - i. 登录Apsara Uni-manager运营控制台。
 - ii. 单击页面右上角的用户头像, 单击**个人信息**。
 - iii. 单击查看当前角色策略。

在当前选择角色策略对话框中查看 RAM Role 。

2. 获取STS Accesskey三元组。

开发指南·<mark>准备工作</mark> 数据管理

? 说明

以下代码示例中可变参数均需替换为您实际的相关信息,尖括号(<>)在最终代码中不保留。涉及替换的可变参数如下:

- <yourRegionID>, 请替换为您实际的地域ID。
- 。 <yourAccessKeyID>, 请替换为您实际的AccessKey ID。
- 。 <yourAccessKeySecret>, 请替换为您实际的AccessKey Secret。
- 。 <yourRoleSessionName>, 请替换为您当前实际的会话名称。
- <yourRoleArn>, 请替换为您实际的RAM Role。
- 。 <sts.aliyuns.com> , 请替换为实际的STS Endpoint 。

数据管理 开发指南·<mark>准备工作</mark>

```
public static void main(String[] args) {
   // 创建DefaultAcsClient实例并初始化。
   // 请替换为您实际的地域ID、AccessKey ID、AccessKey Secret
   DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourAccessKe</pre>
yID>", "<yourAccessKeySecret>");
   HttpClientConfig clientConfig = HttpClientConfig.getDefault();
   clientConfig.setIgnoreSSLCerts(true);
   clientConfig.setProtocolType(ProtocolType.HTTPS);
   profile.setHttpClientConfig(clientConfig);
   IAcsClient client = new DefaultAcsClient(profile);
   String assumeRole = callAssumeRole(client);
   if (StringUtils.isEmpty(assumeRole)) return;
   JSONObject assumeRoleJson = JSONObject.parseObject(assumeRole);
   JSONObject credentials = assumeRoleJson.getJSONObject("Credentials");
   // 获取到STS三元组
   String stsAK = credentials.getString("AccessKeyId");
   String stsSK = credentials.getString("AccessKeySecret");
   String stsToken = credentials.getString("SecurityToken");
private static String callAssumeRole(IAcsClient client) {
   CommonRequest request = new CommonRequest();
   // "sts.aliyuns.com" 请替换为实际的STS Endpoint, 一般格式为sts-vpc.${global:region}.${
global:internet-domain}
   request.setSysDomain("sts.aliyuns.com");
   request.setSysProduct("Sts");
   request.setSysAction("AssumeRole");
   request.setSysVersion("2015-04-01");
   request.setSysMethod(MethodType.POST);
   request.setHttpContentType(FormatType.FORM);
   // 业务参数,设置本次调用Session的别名,"<yourRoleSessionName>"请您替换为实际值
   request.putQueryParameter("RoleSessionName", "<yourRoleSessionName>");
   // 扮演的Role角色ARN,"<yourRoleArn>"请替换为上一步获取到的实际Ram Role ARN
   request.putQueryParameter("RoleArn", "<yourRoleArn>");
   try {
       CommonResponse response = client.getCommonResponse(request);
       return response.getData();
    } catch (ServerException e) {
       System.out.println(e.getErrCode());
       System.out.println(e.getErrMsg());
    } catch (ClientException e) {
       System.out.println(e.getErrCode());
       System.out.println(e.getErrMsg());
   return null;
```

3. 通过STS AccessKey三元组调用云产品的接口,下列代码以ECS的 DescribeZones 接口为示例。

开发指南·<mark>准备工作</mark> 数据管理

? 说明

以下代码示例中可变参数均需替换为您实际的相关信息,尖括号(<>)在最终代码中不保留。涉及替换的可变参数如下:

- <yourRegionID>, 请替换为您实际的地域ID。
- 。 <yourStsAccesskeyID>,请替换为您实际的STS Accesskey ID。
- <yourStsAccessKeySecret>, 请替换为您实际的STS Accesskey Secret。
- <yourStsToken>, 请替换为实际STS三元组中的SecurityToken。
- <yourResourceGroupID>, 请替换为您实际的资源集ID。
- <yourOrganizationID>请替换为您实际的组织ID。

```
public static void main(String[] args) {
   // "<yourRegionID>"请替换为您实际的地域ID
   // "<yourStsAccessKeyID>", "<yourStsAccessKeySecret>"请替换为前一步获取到的STS三元组中A
ccessKeyID, AccessKeySecret
   DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourStsAcces
skey>", "<yourStsAccessSecret>");
   HttpClientConfig clientConfig = HttpClientConfig.getDefault();
   clientConfig.setIgnoreSSLCerts(true);
   clientConfig.setProtocolType(ProtocolType.HTTPS);
   profile.setHttpClientConfig(clientConfig);
   IAcsClient client = new DefaultAcsClient(profile);
   CommonRequest request = new CommonRequest();
   request.setSysDomain("ecs.aliyuns.com");
   request.setSysProduct("Ecs");
   request.setSysAction("DescribeZones");
   request.setSysVersion("2014-05-26");
   request.setSysMethod(MethodType.POST);
   request.setHttpContentType(FormatType.FORM);
   //"<yourRegionID>" 请替换为您实际的地域ID
   request.putHeadParameter("x-acs-regionid", "<yourRegionID>");
   // "<yourStsToken>" 请替换为您前一步获取到的STS三元组中SecurityToken
   request.putQueryParameter("SecurityToken", "<yourStsToken>");
   // 非必填,请替换为您实际的资源集ID
   // request.putHeadParameter("x-acs-resourcegroupid", "<yourResourceGroupID>");
   // 必填,请替换为您实际的组织ID
   request.putHeadParameter("x-acs-organizationid", "<yourOrganizationID>");
   try {
       CommonResponse response = client.getCommonResponse(request);
       System.out.println(response.getData());
    } catch (ServerException e) {
       System.out.println(e.getErrCode());
       System.out.println(e.getErrMsg());
    } catch (ClientException e) {
       System.out.println(e.getErrCode());
       System.out.println(e.getErrMsg());
```

 数据管理 开发指南·准备工作

2.5. 获取DMS的Endpoint

本文介绍了如何获取DMS的Endpoint。

背景信息

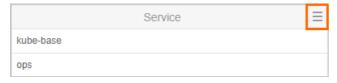
通过POP网关调用专有云API时需要使用云产品业务域的Endpoint。您可以在阿里云驻场运维工程师处获取。您也可以通过如下方法自行获取Endpoint。

操作步骤

- 1. 登录飞天基础运维平台。
- 2. 在左侧导航栏中单击报表。
- 3. 在全部报表页面搜索 服务注册变量 , 单击报表名称服务注册变量。



4. 在服务注册变量页面,单击Service旁的 = 图标,搜索 dts-dtm 。



5. 在Service Regist ration列,单击鼠标右键,选择显示更多。

在详情页面,查看dms_open_api_domain的值,即 Endpoint 。

2.6. 获取DMS的SDK包

通过POP网关调用专有云API时,需使用数据管理DMS的SDK。本文为您介绍如何获取阿里云开发工具包(SDK)。



开发指南·<mark>准备工作</mark> 数据管理

语言	数据管理DMS SDK下载地址	文档
Alibaba Cloud SDK for Python	DMS SDK for Python	快速开始(Python)
Alibaba Cloud SDK for Go	DMS SDK for Go	快速开始(Go)

? 说明

推荐使用 Java、Python、Go三种语言的SDK包,使用其他语言的SDK包可能发生不兼容情况。

3.Java SDK调用示例

3.1. 快速开始

本节介绍如何获取数据管理DMS Java SDK并开始调用。

环境准备

- 使用阿里云专有云Java SDK,您需要一个授权账号以及一对AccessKey ID和AccessKey Secret。参见获取 AccessKey在Apsara Uni-manager运营控制台上获取和查看您的AccessKey。
- 获取数据管理DMS的Endpoint,具体操作参见获取DMS的Endpoint。
- 阿里云专有云Java SDK需要1.8及以上的JDK。

安装阿里云专有云Java SDK

如果您使用Apache Maven来管理Java项目,只需在项目的pom.xml中导入JAR文件。

要使用数据管理DMS的开发工具包,必须安装SDK核心库以及产品的SDK。其中,SDK核心库推荐使用4.5.17及以上版本,SDK的版本为1.18.0。

使用Java SDK

调用阿里云专有云Java SDK的主要操作步骤如下:

- 1. 创建并初始化DefaultAcsClient实例。
- 2. 创建API请求并设置参数。
- 3. 发起请求并处理应答或异常。

? 说明

以下代码示例中可变参数均需替换为您实际的相关信息,尖括号(<>)在最终代码中不保留。

涉及替换的可变参数如下:

- <yourRegionID>, 请替换为您实际的地域ID。
- 。 <yourAccessKeyID>, 请替换为您实际的AccessKey ID。
- 。 <yourAccessKeySecret>, 请替换为您实际的AccessKey Secret。
- <DMSProduct Code>,请替换为您专有云环境DMS的产品Code。
- <DMSProductEndpoint>, 请替换为您专有云环境DMS产品的Endpoint地址。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.dms enterprise.model.v20181101.GetOpLogRequest;
import com.aliyuncs.dms enterprise.model.v20181101.GetOpLogResponse;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
public class GetOpLogSampleTest {
    public static void main(String[] args) {
        String regionId = "<yourRegionID>";
        String accessKey = "<yourAccessKeyID>";
        String secretKey = "<yourAccessKeySecret>";
        String product = "<DMSProductCode>";
        String endpoint = "<DMSProductEndpoint>";
        DefaultProfile.addEndpoint(regionId, product, endpoint);
        DefaultProfile profile = DefaultProfile.getProfile(regionId, accessKey, secretK
ey);
        IAcsClient client = new DefaultAcsClient(profile);
        GetOpLogRequest request = new GetOpLogRequest();
        request.setStartTime("2022-01-01 00:00:00");
        request.setEndTime("2022-01-02 00:00:00");
        request.setPageNumber(1);
        request.setPageSize(100);
        try {
            GetOpLogResponse response = client.getAcsResponse(request);
            System.out.println("RequestId: " + response.getRequestId());
            System.out.println("OpLogCount: " + response.getTotalCount());
        catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
```

3.2. 使用手册

3.2.1. 安装Java SDK

本文介绍如何安装及调用数据管理DMS Java SDK。

 阿里云专有云Java SDK支持1.8及以上版本的JDK,可以通过添加Maven依赖或下载阿里云开发工具包(SDK)的方式安装专有云Java SDK。

要使用数据管理DMS的开发工具包,必须安装阿里云Java SDK核心库以及数据管理DMS的Java SDK。SDK核心库推荐使用4.5.17及以上版本,SDK的版本为1.18.0。

以使用数据管理DMS的Java SDK为例,您需要安装阿里云Java SDK核心库和DMS Java SDK,并在pom.xml文件中添加以下依赖即可。准确的SDK版本号,请参见获取DMS的SDK包。

3.2.2. 设置身份验证凭证

当使用阿里云专有云SDK访问阿里云专有云服务时,您需要提供授权的账号进行身份验证。本文介绍了如何设置身份验证凭证。

目前,Java SDK支持通过AccessKey进行身份验证,获取AccessKey的方法请参见<mark>获取AccessKey</mark>。 使用AccessKey作为访问凭据时,需要在初始化Client 时设置凭证。

? 说明

确保包含AccessKey的代码不会泄漏(例如提交到外部公开的Git Hub项目),否则将会危害您的信息安全。

```
DefaultProfile profile = DefaultProfile.getProfile(
    "<yourRegionID>", // 请替换为您实际的地域ID
    "<yourAccessKeyID>", // 请替换为您实际的AccessKey ID
    "<yourAccessKeySecret>"); // 请替换为您实际的AccessKey Secret
```

? 说明

以上代码示例中可变参数均需替换为您实际的相关信息,尖括号(<>)在最终代码中不保留。

3.2.3. 请求连接配置

本文介绍了如何配置请求连接(HttpClientConfig)。

您可通过以下代码配置HttpClientConfig,具体请根据自身服务能力自行选择配置项并做相应配置。

? 说明

以下代码示例中可变参数均需替换为您实际的相关信息,尖括号(<>)在最终代码中不保留。 涉及替换的可变参数如下:

- <readTimeout>,请您根据自身服务的能力,配置读取超时时间,单位为毫秒(ms)。
- <connectionTimeout>,请您根据自身服务的能力,配置连接超时时间,单位为毫秒(ms)。
- <maxRequest sPerHost>,请您根据自身服务的能力,配置每台服务器的最大请求数,数据类型为INT。
- <maxRequests>,请您根据自身服务的能力,配置最大请求数,数据类型为INT。
- <http://192.168.XX.XX:33XX>, 请替换为您实际的代理IP+端口。

```
// 获取HttpClientConfig对象
HttpClientConfig clientConfig = HttpClientConfig.getDefault();
// 设置是否忽略证书
clientConfig.setIgnoreSSLCerts(true);
// (可选)设置读取超时时间,单位毫秒 (ms)
clientConfig.setReadTimeoutMillis(<readTimeout>);
// (可选)设置创建连接超时时间,单位毫秒 (ms)
clientConfig.setConnectionTimeoutMillis(<connectionTimeout>);
// (可选)设置每个主机的最大请求数,数据类型为INT
clientConfig.setMaxRequestsPerHost(<maxRequestsPerHost>);
// (可选)设置最大请求数,数据类型为INT
clientConfig.setMaxRequests(<maxRequests>);
// (可选)设置协议类型
clientConfig.setProtocolType(ProtocolType.HTTP);
/** (可选)请根据实际情况判断是否设置代理,设置方法如下:
clientConfig.setHttpProxy("<http://192.168.XX.XX:33XX>");
clientConfig.setHttpsProxy("<https://192.168.XX.XX:33XX>");
请替换为您实际的代理IP+端口
profile.setHttpClientConfig(clientConfig);
```

3.2.4. 发起调用

本文介绍如何使用数据管理DMS Java SDK发起请求。

操作步骤

1. 新建一个AcsClient。

```
IAcsClient client = new DefaultAcsClient(profile);
```

2. 创建请求。

```
请求类的命名规范为 ${apiName}Request , 其中${apiName}为API名称,例如GetOpLog。在引入多个产品SDK时,有可能存在Request类同名的情况,请注意按照package区分。
```

```
GetOpLogRequest request = new GetOpLogRequest();
request.setStartTime("2022-01-01 00:00:00");
request.setEndTime("2022-01-02 00:00:00");
request.setPageNumber(1);
request.setPageSize(100);
```

3. 发起调用并处理应答。

```
GetOpLogResponse response = client.getAcsResponse(request);
```

正常情况下,应答中的所有字段,都会被反序列化到response中,您可以直接调用 response.getXXX() 来获得应答中的字段。

```
OpLogCount := response.getTotalCount()
```

如果出现了异常,或您需要原始HTTP应答的情况下,您可以通过使用 doAction() 来获取原始应答。

```
HttpResponse response = client.doAction(request);
```

3.2.5. 错误处理

在使用Java SDK时,当服务端或者SDK端出错时,SDK会返回相应的异常信息。这些异常信息中会包含具体的错误信息包括错误码(Error Code)和错误消息(Error Message)。

一般情况下,您不需要Java SDK返回的错误进行处理。您只需要根据服务端的错误提示处理服务端错误即可。

- ServerException 是服务端错误信息。
- ClientException 是SDK端错误信息。

例如,当出现如下错误时,您可以根据具体的错误信息,修改AccessKey的ID即可。

```
com.aliyuncs.exceptions.ClientException: InvalidAccessKeyId.NotFound : Specified access key
is not found.
```

如果您需要对SDK的错误进行处理,可以参考如下示例编写代码:

```
try {
   FooResponse response = client.getAcsResponse(request);
   // 处理Response
   // ...
}catch (ServerException e) {
   // 这里可以添加您自己的错误处理逻辑
   // 例如,打印具体的错误信息
   System.out.println("ErrorCode=" + e.getErrCode());
   System.out.println("ErrorMessage=" + e.getErrMsg());
   // 如果遇上棘手问题,可以提交工单,并将RequestId提供给我们
   System.out.println("ResponseId=" + e.getRequestId());
}catch (ClientException e) {
   // 这里可以添加您自己的错误处理逻辑
   // 例如,打印具体的错误信息
   System.out.println("ErrorCode=" + e.getErrCode());
   System.out.println("ErrorMessage=" + e.getErrMsg());
```

4.Python SDK调用示例

4.1. 快速开始

本节介绍如何获取数据管理DMS的Python SDK并开始调用。

环境准备

- 使用Python SDK前,您需要一个授权账号以及一对AccessKey ID和AccessKey Secret。具体请参见<mark>获取AccessKey。</mark>
- 获取数据管理DMS的Endpoint,具体操作参见获取DMS的Endpoint。

安装Python SDK

- 1. 安装SDK核心库。
 - 如果您使用的是Python 2.x, 执行以下命令, 安装阿里云SDK核心库:

pip install aliyun-python-sdk-core

○ 如果您使用的是Python 3.x, 执行以下命令, 安装阿里云SDK核心库:

pip3 install aliyun-python-sdk-core-v3

2. 安装SDK。

以下以安装数据管理DMS的SDK为例:

○ 如果您使用的是Python 2.x, 执行以下命令, 安装DMS的SDK:

pip install aliyun-python-sdk-dms-enterprise

○ 如果您使用的是Python 3.x, 执行以下命令, 安装DMS的SDK:

pip3 install aliyun-python-sdk-dms-enterprise

使用Python SDK

以下示例以数据管理DMS为例展示调用Python SDK的3个主要步骤:

- 1. 创建Client实例。在创建Client实例时,您需要获取Region ID、AccessKey ID和AccessKey Secret。
- 2. 创建API请求并设置参数。
- 3. 发起请求并处理应答或异常。

4.2. 使用手册

4.2.1. 安装Python SDK

本文档介绍如何安装及调用数据管理DMS Python SDK。

阿里云专有云Python SDK支持Python 2.6.x、2.7.x和3.x及以上环境,您可以通过pip安装专有云Python SDK。

- ② 说明 使用产品的开发工具包,都必须安装阿里云Python SDK核心库。例如,如果要使用DMS的Python SDK,您需要安装阿里云Python SDK核心库和DMS的Python SDK。
- 如果您使用的是Python 2.x, 执行以下命令, 安装阿里云SDK核心库:

```
pip install aliyun-python-sdk-core
```

● 如果您使用的是Python 3.x, 执行以下命令, 安装阿里云SDK核心库:

```
pip3 install aliyun-python-sdk-core-v3
```

安装云产品的SDK,以下以安装DMS的SDK为例。

4.2.2. 设置身份验证凭证

当使用阿里云SDK访问阿里云专有云服务时,您需要提供授权的账号进行身份验证。

目前Python SDK支持通过AccessKey进行身份验证,获取AccessKey的方法请参见获取AccessKey。 使用AccessKey作为访问凭据时,需要在初始化Client 时设置凭证。

□ 注意 确保包含AccessKey的代码不会泄漏(例如提交到外部公开的Git Hub项目),否则将会危害您的信息安全。

```
clt = client.AcsClient(secret id, secret key, region id)
```

4.2.3. 设置HTTPS请求

Python SDK支持使用HTTP和HTTPS协议发起 API 请求。

使用Python SDK时,您可以为某个请求指定使用HTTP或HTTPS协议,您也可以设置全局默认协议。

② 说明 产品的默认协议(HTTP/HTTPS)优先于设置的全局默认协议。

添加OpenSSL依赖

Python SDK的HTTPS协议依赖Python的OpenSSL支持。要使用阿里云SDK通过HTTPS协议发送请求,您需要在Python中添加OpenSSL支持。Open SSL支持TLS1.2及以上版本。

运行 python -c "import ssl" 检查Python环境是否支持OpenSSL。运行后,如果没有出现

ImportError: No module named ssl 的错误信息,说明已经支持OpenSSL。

若没有OpenSSL,运行以下命令安装:

```
pip install pyopenssl
```

设置单个请求的HTTP/HTTPS协议

参考以下代码示例为一个接口设置HTTPS调用:

```
request = DescribeInstancesRequest()
request.set_protocol_type("https")
# 取值: "https" 或 "http"
```

设置全局默认协议

参考以下代码示例设置全局默认协议:

```
import aliyunsdkcore.request
aliyunsdkcore.request.set_default_protocol_type("https")
# 创建请求并调用 client.do_action_with_exception() 来发送请求
```

4.2.4. 发起调用

本节介绍如何使用数据管理DMS Python SDK发起请求。

操作步骤

1. 导入DMS的SDK。

```
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkcore.auth.credentials import AccessKeyCredential
from aliyunsdkcore.auth.credentials import StsTokenCredential
from aliyunsdkdms_enterprise.request.v20181101.GetOpLogRequest import GetOpLogRequest
```

2. 新建一个AcsClient。

```
credentials = AccessKeyCredential('<yourAccessKeyID>', '<yourAccessKeySecret>')
client = AcsClient(region_id='<yourRegionID>', credential=credentials)
```

3. 创建Request对象。

```
request = GetOpLogRequest()
request.set_accept_format('json')
request.set_endpoint('<DMSProductEndpoint>')
```

4. 发起调用并处理返回。

```
response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding='utf-8'))
response = clt.do_action_with_exception(request)
print response
```

4.2.5. 错误处理

在使用Python SDK时,当服务端或者SDK端出错时,SDK会返回相应的异常信息。这些异常信息中会包含具体的错误信息包括错误码(Error Code)和错误消息(Error Message)。

一般情况下,您不需要对Python SDK返回的错误进行处理。您只需要根据服务端的错误提示处理服务端错误即可。

- ServerException 是服务端错误信息。
- ClientException 是SDK端错误信息。

例如,当出现如下错误时,您可以根据具体的错误信息,修改AccessKey的ID即可。

aliyunsdkcore.acs_exception.exceptions.ServerException: HTTP Status: 404 Error:InvalidAcce ssKeyId.NotFound Specified access key is not found.

如果您需要对SDK的错误进行处理,可以参考如下示例编写代码:

```
try:
    response = client.do_action_with_exception(request)
except ServerException as e:
    # 这里可以添加您自己的错误处理逻辑
    # 例如,打印具体的错误信息
    print e.get_http_status()
    print e.get_error_code()
    print e.get_error_msg()
```

5.Go SDK调用示例

5.1. 快速开始

阿里云Go SDK让您不用复杂编程即可访问。本操作介绍如何获取阿里云数据管理DMS Go SDK并开始调用。

环境准备

- 使用Python SDK前,您需要一个授权账号以及一对AccessKey ID和AccessKey Secret。 参见<mark>获取 AccessKey</mark>在Apsara Uni-manager运营控制台上获取和查看您的AccessKey。
- 获取数据管理DMS的Endpoint,具体操作参见获取DMS的Endpoint。

安装阿里云Go SDK核心包

阿里云Go SDK支持Go 1.10.x及更高版本,您可以通过执行以下命令安装Go SDK核心包。

go get -u github.com/aliyun/alibaba-cloud-sdk-go/sdk

使用Go SDK

以下这个代码示例展示了调用阿里云Go SDK的3个主要步骤:

- 1. 初始化Client实例。
- 2. 创建API请求并设置参数。
- 3. 发起请求并处理应答或异常。
 - ② 说明 以下代码示例中可变参数均需替换为您实际的相关信息,尖括号(<>)在最终代码中不保留。

涉及替换的可变参数如下:

- <yourRegionID>, 请替换为您实际的地域ID。
- 。 <yourAccessKeyID>, 请替换为您实际的AccessKey ID。
- 。 <yourAccessKeySecret>, 请替换为您实际的AccessKey Secret。
- 。 <DMSProduct Code>, 请替换为您专有云环境DMS的产品Code。
- 。 <DMSProductEndpoint>, 请替换为您专有云环境DMS产品的Endpoint地址。

开发指南·GO SDK调用示例 数据管理

```
package main
import (
   "fmt"
    dms_enterprise "github.com/aliyun/alibaba-cloud-sdk-go/services/dms-enterprise"
)
func main() {
    client, err := dms_enterprise.NewClientWithAccessKey("<yourRegionID>", "<yourAccessKey
ID>", "<yourAccessKeySecret>")
    request := dms_enterprise.CreateGetOpLogRequest()
    request.Scheme = "https"
        request.Domain = "<DMSProductEndpoint>"
    response, err := client.GetOpLog(request)
    if err != nil {
        fmt.Print(err.Error())
    }
    fmt.Printf("response is %#v\n", response)
}
```

4. (可选)切换HTTP/HTTPS协议。

```
request.Scheme = "HTTPS"
```

5.2. 使用手册

5.2.1. 安装Go SDK核心包

本文档介绍如何安装 Go SDK核心包。

阿里云Go SDK支持Go 1.10.x及以上版本,您可以通过执行以下命令安装Go SDK核心包。

```
go get -u github.com/aliyun/alibaba-cloud-sdk-go/sdk
```

5.2.2. 设置身份验证凭证

当使用阿里云专有云SDK访问阿里云专有云服务时,您需要提供授权的账号进行身份验证。

目前Go SDK支持通过AccessKey进行身份验证,获取AccessKey的方法请参见获取AccessKey。

使用AccessKey作为访问凭据,需要在初始化Client时设置凭证。

□ 注意 确保包含AccessKey的代码不会泄漏(例如提交到外部公开的Git Hub项目),否则将会危害您的阿里云账号的信息安全。

以下代码以DMS为例,展示初始化Client实例。

② 说明 以上代码示例中可变参数均需替换为您实际的相关信息,尖括号(<>)在最终代码中不保留。

5.2.3. 发起调用

本文介绍如何使用阿里云数据管理DMS Go SDK发起调用。

以下步骤以数据管理DMS为例,展示使用DMS Go SDK发起调用。

- 1. 导入数据管理DMS SDK。
- 2. 新建一个产品名Client,该Client中包含产品名的所有API。
 - ② 说明 以下代码示例中可变参数均需替换为您实际的相关信息,尖括号(<>)在最终代码中不保留。

```
Client,err := dms_enterprise.NewClientWithAccessKey( //请替换为您实际的产品名称
"<yourRegionID>", // 请替换为您实际的RegionID
"<yourAccessKeyID>", // 请替换为您实际的AccessKeyID
"<yourAccessKeySecret>") // 请替换为您实际的AccessKeySecret
```

3. 使用 xxx.CreateXXXRequest 方法创建一个请求。

该方法的命名规范为 \${service}.Create\${apiName}Request , 其中:

- \${service}为产品名称(小写)。例如 dms_ent erprise。
- \${apiName}为API名称,例如GetOpLog。
- 4. 发起调用。

```
response, err := client.GetOpLog(request)
```

5.2.4. 错误处理

阿里云Go SDK在服务端或者SDK端出错时,会返回相应的错误信息,即 sdk.errors 包下的 ClientError 和 ServerError 。这些错误是基于Go的标准Error接口实现的,您可以像处理Go标准错误一样来处理阿里云Go SDK返回的错误。

简单处理异常

您可以通过以下代码来简单处理错误。

```
response,err := Client.DescribeInstances(request)
if err != nil {
    //获取异常信息
    err.Error()
}
```

ClientError

当SDK内部的任意调用出现error,并无法自动化处理时,SDK会返回一个 ClientError 。

以下以DMS为例,您可以通过以下代码来判断并获取 ClientError 中的错误信息:

开发指南·Go SDK调用示例 数据管理

ServerError

当服务端返回错误的应答时,SDK会将该应答封装成 ServerError 返回。在这种情况下,您依然可以从 Response中获得原始的HTTP应答。

以下以DMS为例,您可以通过以下代码来判断并获取ServerError以及Response中的错误信息:

数据管理 开发指南·API参考

6.API参考 6.1. API概览

数据管理DMS提供以下API接口。

用户管理

API	描述
RegisterUser	录入新的用户。
DisableUser	禁用用户。
EnableUser	启用用户。
UpdateUser	更新用户信息。
DeleteUser	删除用户。
ListUsers	获取用户详情列表。
ListUserTenants	获取当前的租户列表。
GetUserActiveTenant	获取当前租户信息。
GetUser	获取用户信息。

实例管理

API	描述
RegisterInstance	录入新的数据库实例。
ListInstances	进行实例列表检索并获取实例详情。
GetInstance	获取实例详细信息。
UpdateInstance	更新实例信息,检查连通性。
DeleteInstance	删除DMS管理的实例。

操作日志

API	描述
Get OpLog	获取指定时间内的操作日志详情列表。

审批流程

开发指南·API参考 数据管理

API	描述
ListWorkFlowTemplates	获取审批模板列表。
ListWorkFlowNodes	获取审批节点列表。

库表元数据

API	描述
SearchDatabase	搜索数据库的详情信息。
SearchT able	搜索表详情列表
ListDatabases	获取实例下的数据列表详情信息。
ListLogicDatabases	获取逻辑库详情列表。
ListTables	获取数据库下面的表详情列表。
ListLogicTables	获取指定逻辑库下逻辑表的详情列表。
ListColumns	获取表的列信息。
ListIndexes	获取表的索引信息。
GetDatabase	获取数据库详情信息。
GetLogicDatabase	获取逻辑库的详情信息。
Get Met aT able Det ail Info	获取表的字段、索引等信息。
GetTableDBTopology	获取数据表的拓扑图。
Get Met aT ableColumn	获取表的字段信息。
SyncDatabaseMeta	同步数据库字典元数据。
SyncInstanceMeta	同步实例下所有数据库字典元数据。

工单操作

分类	API	描述
通用	ApproveOrder	处理工单审批。
	CloseOrder	关闭工单。
	GetApprovalDetail	获取工单审批详情。
	GetOrderBaseInfo	获取工单基本信息。
	ListOrders	获取工单列表信息。

分类	API	描述	
	Submit Order Approval	提交工单审批。	
	CreateOrder	创建工单。	
	ExecuteDataCorrect	执行数据变更工单。	
数据变更	GetDataCorrectBackupFiles	获取数据变更备份文件附件下载地址。	
	Get DataCorrect Order Detail	获取数据变更工单详情信息。	
结构设计	CreatePublishGroupTask	执行结构设计变更。	
	ExecuteDataExport	执行数据导出工单。	
数据导出	GetDataExportDownloadURL	获取数据导出结果附件下载地址。	
	Get Dat a Export Order Detail	获取数据导出工单详情。	

权限管理

API	描述
RevokeUserPermission	回收用户权限。
ListDatabaseUserPermssions	获取指定数据库权限的用户权限详情。
SetOwners	设置库表Owner。
ListUserPermissions	展示指定用户所具备的库表权限。
GrantUserPermission	授权用户库表权限。

安全管理

API	描述
ListSensitiveColumns	搜索敏感字段列表。
ListSensitiveColumnsDetail	获取敏感字段详情信息。

数据查询

API	描述
ExecuteScript	执行SQL脚本。

6.2. 用户管理

6.2.1. RegisterUser

调用RegisterUser接口录入本企业的新用户。

DMS企业版用户的**管理员**(即产品内【系统管理】-【用户管理】中的管理员角色的用户)都可以通过这个接口增加录入本企业的新用户。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	RegisterUser	系统规定参数。取值:RegisterUser。
Tid	Long	是	-1	租户ID。
Uid	String	是	12345678	需要录入用户的阿里云UID。
UserNick	String	否	dmstest	用户名称。
RoleNames	String	否	USER,DBA	用户角色,取值和说明如下: USER: 普通用户 DBA: DBA ADMIN: 管理员 SECURITY_ADMIN: 安全管理员 问说明 为空则初始化为"普通用户",可选"普通用户"、"安全管理员"、"安全管理员"四个的任意组合,组合多个角色,传递时用*英文逗号*拼接。
Mobile	String	否	130000000xx	联系电话。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	403	错误码。
ErrorMessage	String	The specified user already exists.	错误消息。

数据管理 开发指南· API参考

名称	类型	示例值	描述
RequestId	String	34E01EDD-6A16- 4CF0-9541- C644D1BE01AA	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=RegisterUser
&Tid=-1
&Uid=12345678
&UserNick=dmstest
&<公共请求参数>
```

正常返回示例

XML 格式

JSON 格式

```
{
    "RequestId": "34E01EDD-6A16-4CF0-9541-C644D1BE01AA",
    "Success": true
}
```

6.2.2. DisableUser

调用DisableUser接口禁用云账号。

此OpenAPI为产品内【系统管理】-【用户管理】内的"禁用用户"功能,支持DMS企业版内的"管理员"角色用户通过调用API的方式对暂时不需要的云账号进行产品内的禁用操作。禁用后,对应云账号开通的数据源权限、数据owner配置、DBA配置均会回收与失效。

② 说明 禁用只是临时限制该云账号不能登录本企业的DMS企业版产品,并不是禁用云账号本身。禁用的效果是用户通过该账号不再能登录到DMS企业版(除非再次启用),此时仍占用企业开通规格。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DisableUser	系统规定参数。取值:DisableUser。
Tid	Long	是	-1	租户ID。
Uid	String	是	12345	阿里云账号UID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	403	错误码。
ErrorMessage	String	The specified user not exists.	错误消息。
RequestId	String	34E01EDD-6A16- 4CF0-9541- C644D1BE01AA	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=DisableUser
&Tid=-1
&Uid=12345

&<公共请求参数>

正常返回示例

XML 格式

<DisableUserResponse>

<RequestId>ADE24A7C-DB8B-4D32-94AD-E50E3B921197/RequestId>

<Success>true</Success>

</DisableUserResponse>

JSON 格式

数据管理 开发指南·API参考

```
{
    "RequestId":"34E01EDD-6A16-4CF0-9541-C644D1BE01AA",
    "Success":true
}
```

6.2.3. EnableUser

调用EnableUser接口启用被禁用的云账号。

此OpenAPI为产品内【系统管理】-【用户管理】内的"启用用户"功能,支持DMS企业版内的"管理员"角色用户,通过调用API的方式对先前禁用的云账号进行产品内的启用操作。启用后,对应云账号可恢复产品的登录和相关使用。

② **说明** 启用只是解除该云账号之前禁用不能登录本企业的DMS企业版产品的限制,非云账号本身的 其他操作。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	EnableUser	系统规定参数。取值:EnableUser。
Tid	Long	是	-1	租户ID。
Uid	String	是	12345	阿里云账号UID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	403	错误码。
ErrorMessage	String	The specified user not exists.	错误消息。
RequestId	String	34E01EDD-6A16- 4CF0-9541- C644D1BE01AA	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=EnableUser
&Tid=-1
&Uid=12345
&<公共请求参数>
```

正常返回示例

XML 格式

```
<EnableUserResponse>
    <RequestId>ADE24A7C-DB8B-4D32-94AD-E50E3B921197</RequestId>
        <Success>true</Success>
</EnableUserResponse>
```

JSON 格式

```
{
    "RequestId": "34E01EDD-6A16-4CF0-9541-C644D1BE01AA",
    "Success": true
}
```

6.2.4. UpdateUser

调用UpdateUser接口更新用户信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateUser	系统规定参数。取值:UpdateUser。
Tid	Long	是	-1	租户ID。
Uid	Long	是	123456789	待更新的用户UID。
UserNick	String	否	test	用户名称。
RoleNames	String	否	ADMIN,DBA	用户角色列表,角色取值参 考RegisterUser接口。
Mobile	String	否	188xxxxxxxx	用户钉钉号(手机号码)。
MaxExecut eCount	Long	否	1000	当天查询上限次数。

名称	类型	是否必选	示例值	描述
MaxResult Count	Long	否	1000	当天查询上限行数。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	E9BEBF41-4F69- 4605-A5D5- A67955173941	API请求唯一ID。
Success	Boolean	false	更新用户是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=UpdateUser
&Tid=-1
&Uid=123456789
&UserNick=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>E9BEBF41-4F69-4605-A5D5-A67955173941</RequestId>
<Success>true</Success>
```

JSON 格式

```
{
   "RequestId": "E9BEBF41-4F69-4605-A5D5-A67955173941",
   "Success": true
}
```

6.2.5. DeleteUser

调用DeleteUser接口删除云账号。

此OpenAPI为产品内【系统管理】-【用户管理】内的"删除用户"功能,支持DMS企业版内的"管理员"角色用户通过调用API的方式对不再需要的云账号进行产品内的删除操作。删除后,对应云账号开通的数据源权限、数据owner配置、DBA配置均会回收与失效。

② 说明 删除只是去除与本企业的DMS企业版产品内的用户的关联关系,并不是删除云账号本身。删除的效果是用户通过该账号不再能登录到DMS企业版(除非再次加入)。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteUser	系统规定参数。取值:DeleteUser。
Tid	Long	是	-1	租户ID。
Uid	String	是	12345	阿里云账号UID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	403	错误码。
ErrorMessage	String	The specified user not exists.	错误消息。
RequestId	String	34E01EDD-6A16- 4CF0-9541- C644D1BE01AA	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=DeleteUser
&Tid=-1
&Uid=12345

&<公共请求参数>

正常返回示例

XML 格式

JSON 格式

```
{
    "RequestId":"34E01EDD-6A16-4CF0-9541-C644D1BE01AA",
    "Success":true
}
```

6.2.6. ListUsers

调用ListUsers接口获取用户详情列表。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListUsers	系统规定参数。取值:ListUsers。
Tid	Long	是	-1	租户ID。
Role	String	否	DBA	角色,取值为参考 RegisterUser 接口。
UserState	String	否	NORMAL	用户状态,取值和说明如下: NORMAL: 正常 DISABLE: 禁用 DELETE: 已删除
SearchKey	String	否	test	搜索关键字。
PageNumber	Integer	否	1	页码。
PageSize	Integer	否	50	每页记录数。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	98D35416-2B92- 4CE5-8FD2- B1E61E165536	请求ID。
Success	Boolean	true	请求是否成功。
TotalCount	Long	1	满足条件的记录总数。
UserList	Array of User		用户详情列表。
User			
CurExecuteCoun t	Long	0	当天查询次数。
CurResult Count	Long	0	当天查询行数。
LastLoginTime	String	2020-02-02 00:00:00	最后登录时间。
MaxExecuteCou nt	Long	1000	当天查询次数上限。
MaxResult Coun t	Long	1000	当天查询行数上限。
Mobile	String	188xxxxx	钉钉号(手机号)。
NickName	String	test	名称。
Parent Uid	String	143214352	父账号UID。
RoleIdList	List	1	角色ID列表。
RoleNameList	List	ADMIN	角色列表。

名称	类型	示例值	描述
State	String	NORMAL	用户状态。
Uid	String	14324321533	用户阿里云UID。
Userld	String	1	用户ID。

示例

请求示例

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

&Tid=-1

&<公共请求参数>

正常返回示例

XML 格式

```
<UserList>
   <User>
       <RoleIdList>
           <RoleIds>3</RoleIds>
           <RoleIds>1</RoleIds>
       </RoleIdList>
       <State>NORMAL</State>
       <RoleNameList>
           <RoleNames>ADMIN</RoleNames>
           <RoleNames>USER</RoleNames>
       </RoleNameList>
       <NickName>xxx</NickName>
       <UserId>2</UserId>
       <Uid>111</Uid>
    </User>
   <User>
       <RoleIdList>
           <RoleIds>1</RoleIds>
           <RoleIds>3</RoleIds>
       </RoleIdList>
       <State>NORMAL</State>
       <RoleNameList>
           <RoleNames>USER</RoleNames>
           <RoleNames>ADMIN</RoleNames>
       </RoleNameList>
       <NickName>xxx[子账号]</NickName>
       <ParentUid>111</ParentUid>
       <UserId>1</UserId>
       <Uid>111</Uid>
   </User>
</UserList>
<TotalCount>2</TotalCount>
<RequestId>2D346B20-BF4A-4C96-8295-DE8EB9BF5B90</RequestId>
<Success>true</Success>
```

JSON 格式

数据管理 开发指南·API参考

```
"UserList": {
  "User": [
      "RoleIdList": {
       "RoleIds": [
        3,
        1
       ]
      },
      "State": "NORMAL",
      "RoleNameList": {
       "RoleNames": [
         "ADMIN",
         "USER"
       ]
      "NickName": "xxx",
      "UserId": 2,
      "Uid": "111"
    },
      "RoleIdList": {
       "RoleIds": [
        1,
         3
       ]
      "State": "NORMAL",
      "RoleNameList": {
       "RoleNames": [
        "USER",
        "ADMIN"
      ]
      "NickName": "xxx[子账号]",
      "ParentUid": "111",
      "UserId": 1,
     "Uid": "111"
    }
 ]
},
"TotalCount": 2,
"RequestId": "2D346B20-BF4A-4C96-8295-DE8EB9BF5B90",
"Success": true
```

6.2.7. ListUserTenants

调用ListUserTenants接口获取当前的租户列表。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	List User Tenants	系统规定参数。取 值:ListUserTenants。
RegionId	String	是	cn-hangzhou	选择DMS的服务地域,用于调用该接口。 ② 说明 请根据应用所在地就近选择。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	CA1F4A4D-EE49- 472F-8DF9- F537101E263A	请求ID。
Success	Boolean	true	请求是否成功。
TenantList	Array of Tenant		租户信息列表。
Status	String	ACTIVE	租户状态。取值: ◆ ACTIVE: 活跃,表示用户正处于当前租户下。 ◆ IN_ACTIVE: 非活跃。
TenantName	String	test	租户名。
Tid	Long	-1	租户ID。

示例

请求示例

http(s)://[Endpoint]/?Action=ListUserTenants &RegionId=cn-hangzhou &<公共请求参数>

正常返回示例

数据管理 开发指南·API参考

XML 格式

```
<RequestId>CA1F4A4D-EE49-472F-8DF9-F537101E263A/RequestId>
<TenantList>
   <Status>ACTIVE</Status>
   <TenantName>test1</TenantName>
   <Tid>-1</Tid>
</TenantList>
<TenantList>
   <Status>IN ACTIVE</Status>
   <TenantName>test2</TenantName>
   <Tid>-2</Tid>
</TenantList>
<TenantList>
   <Status>IN ACTIVE</Status>
   <TenantName>tes</TenantName>
   <Tid>-3</Tid>
</TenantList>
```

JSON 格式

6.2.8. GetUserActiveTenant

调用GetUserActiveTenant接口获取当前所处的租户信息。

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	GetUserActiveTen ant	系统规定参数。取 值:GetUserActiveTenant。
				选择DMS的服务地域,用于调用该接口。
RegionId	String	否	cn-hangzhou	说明 请根据应用所在地就近选择。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	E20EAB9F-A3DB- 42EC-BD0C- 102A886340F5	请求ID。
Success	Boolean	true	请求是否成功。
Tenant	Struct		租户信息。
Status	String	ACTIVE	租户状态。取值: ◆ ACTIVE: 活跃,表示用户正处于当前租户下。 ◆ IN_ACTIVE: 非活跃。
TenantName	String	test	租户名。
Tid	Long	-1	租户ID。

示例

请求示例

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

正常返回示例

XML 格式

JSON 格式

```
"Tenant": {
    "Status": "ACTIVE",
    "TenantName": "test",
    "Tid": -1
},
    "RequestId": "E20EAB9F-A3DB-42EC-BD0C-102A886340F5",
    "Success": true
}
```

6.2.9. GetUser

调用GetUser接口获取用户信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetUser	系统规定参数。取值: GetUser 。
Tid	Long	是	-1	租户ID。
Uid	String	否	123344235	阿里云UID。
Userld	String	否	1343	用户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。

名称	类型	示例值	描述
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	CFF30064-5985- 4ACB-B3F9- 9557596913C8	请求ID。
Success	Boolean	true	请求是否成功。
User	Struct		用户信息。
CurExecuteCoun t	Long	0	当天查询次数。
CurResultCount	Long	0	当天查询行数。
LastLoginTime	String	2020-02-02 00:00:00	最后登录时间。
MaxExecuteCou nt	Long	1000	当天查询次数上限。
MaxResult Coun t	Long	1000	当天查询行数上限。
Mobile	String	138xxxxxxxx	用户手机号码。
NickName	String	nickName	用户名称。
Parent Uid	Long	12345352	主账号阿里云Uid。
RoleIdList	List	1	角色ID列表。
RoleNameList	List	普通用户	角色列表。
State	String	NORMAL	用户状态。
Uid	String	1243215435	用户阿里云UID。
Userld	String	122	用户ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetUser
&Tid=-1
&<公共请求参数>
```

正常返回示例

XML 格式

```
<User>
   <RoleIdList>
      <RoleIds>1</RoleIds>
       <RoleIds>4</RoleIds>
       <RoleIds>3</RoleIds>
       <RoleIds>2</RoleIds>
   </RoleIdList>
   <State>NORMAL</State>
   <RoleNameList>
       <RoleNames>普通用户</RoleNames>
       <RoleNames>安全管理员</RoleNames>
       <RoleNames>管理员</RoleNames>
       <RoleNames>DBA</RoleNames>
   </RoleNameList>
   <NickName>xxx</NickName>
   <ParentUid>114324325</ParentUid>
   <UserId>432143</UserId>
   <Uid>41234234324</Uid>
</User>
<RequestId>CFF30064-5985-4ACB-B3F9-9557596913C8/RequestId>
<Success>true</Success>
```

JSON 格式

```
"User": {
 "RoleIdList": {
   "RoleIds": [
     1,
     4,
     3,
     2
   ]
  "State": "NORMAL",
 "RoleNameList": {
   "RoleNames": [
     "普通用户",
     "安全管理员",
     "管理员",
     "DBA"
  ]
 "NickName": "xxx",
 "ParentUid": "114324325",
 "UserId": 432143,
 "Uid": "41234234324"
"RequestId": "CFF30064-5985-4ACB-B3F9-9557596913C8",
"Success": true
```

6.3. 实例管理

6.3.1. RegisterInstance

调用RegisterInstance接口录入本企业新数据库实例。

DMS企业版用户的**管理员、DBA**(即产品内【系统管理】-【用户管理】中的管理员与DBA角色的用户)都可以通过这个接口增加录入本企业的新数据库实例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	RegisterInstance	系统规定参数。取 值:RegisterInstance。
Tid	Long	是	-1	租户ID。

数据管理 开发指南·API参考

名称	类型	是否必选	示例值	描述
InstanceType	String	是	MySQL	数据库类型,取值为: MySQL SQLServer PostgreSQL Oracle DRDS OceanBase Mongo Redis
InstanceSource	String	是	RDS	数据库实例来源,取值和说明如下: • PUBLIC_OWN: 代表公网自建实例 • RDS: 代表RDS实例 • ECS_OWN: 代表ECS自建实例 • VPC_IDC: 代表VPC IDC自建实例
NetworkType	String	是	VPC	网络类型。
EnvType	String	是	product	环境类型,取值和说明如下: product:生产环境 dev:开发环境 pre:预发环境 test:测试环境 sit:SIT环境 uat:UAT环境 pet:压测环境 stag:STAG环境
Host	String	是	dmstest.rds.aliyu n.com	目标数据库的主机地址。
Port	Integer	是	3306	目标数据库的访问端口。
DatabaseUser	String	是	dmstest	数据库访问账号。
DatabasePasswor d	String	是	password	数据库访问密码。
InstanceAlias	String	是	测试实例	实例名称,后续帮助用户快速区分定位。

名称	类型	是否必选	示例值	描述
DbaUid	Long	是	12345678	实例的DBA,传入DBA的阿里云UID。
SafeRule	String	是	测试使用	实例的安全规则,传入企业内的安全规则 名称。
QueryTimeout	Integer	是	60	查询超时时间,单位: s (秒)。
ExportTimeout	Integer	是	600	导出超时时间,单位: s (秒)。
EcsInstanceId	String	否	i-xxxxxxxxx	ECS的实例ID。 ② 说明 InstanceSource取值 为ECS自建库此值必传。
Vpcld	String	否	vpc- xxxxxxxxxxxxxxxxxxxxx xxxxx	VPC ID。 ③ 说明 InstanceSource取值 为VPC专线IDC时此值必传。
EcsRegion	String	否	cn-hangzhou	实例所在区域。 ② 说明 InstanceSource取值 为RDS、ECS自建库、VPC专线IDC时此值必传。
Sid	String	否	dmstest	SID/数据库。 ③ 说明 InstanceType取值为 PostgreSQL、Oracle时此值必传。
DataLinkName	String	否	dblink_test	跨库查询DataLink名称。
DdlOnline	Integer	否	2	【重点关注】是否使用online服务,目前 仅支持MySQL和PolarDB。 ● 0:不使用 ● 1:原生onlineDDL优先 ● 2:DMS无锁表结构变更优先

数据管理 开发指南·API参考

名称	类型	是否必选	示例值	描述
UseDsql	Integer	否	1	是否启用跨实例查询,取值: • 0: 不开启 • 1: 开启
SkipTest	Boolean	否	true	是否跳过测试连接,取值: • true: 跳过测试连接。 • false: 不跳过测试连接。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	404	错误码。
ErrorMessage	String	The specified instance already exists.	错误消息。
RequestId	String	F4E2A94B-604F- 43FF-93E7- F4EE3DCF412E	请求唯一ID。
Success	Boolean	true	● true: 录入成功 ● false: 录入失败

示例

请求示例

```
http(s)://[Endpoint]/?Action=RegisterInstance
&Tid=-1
&InstanceType=MySQL
&InstanceSource=RDS
&NetworkType=VPC
&EnvType=product
&Host=dmstest.rds.aliyun.com
&Port=3306
&DatabaseUser=dmstest
&DatabasePassword=password
&InstanceAlias=测试实例
&DbaUid=12345678
&SafeRule=测试使用
&QueryTimeout=60
&ExportTimeout=600
&<公共请求参数>
```

正常返回示例

XML 格式

```
JSON 格式
```

```
{
    "RequestId": "F4E2A94B-604F-43FF-93E7-F4EE3DCF412E",
    "Success": true
}
```

Python sdk示例

示例为VPC专线实例的录入。

数据管理 开发指南· API参考

```
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkdms enterprise.request.v20181101.RegisterInstanceRequest import RegisterInsta
nceRequest
client = AcsClient(
   "<your-access-key-id>",
   "<your-access-key-secret>",
   "<your-region-id>"
# VPC IDC实例注册示例
request = RegisterInstanceRequest()
# 租户ID
request.set Tid(1)
# 实例host
request.set Host("192.168.16.113")
# 实例port
request.set Port(3314)
# 实例db类型 eg : MySQL
request.set InstanceType("MySQL")
# 实例来源 eg : VPC IDC
request.set InstanceSource("VPC IDC")
# 网络类型 type: eg : VPC
request.set NetworkType("VPC")
request.set EnvType("PRODUCT")
# request.set EcsInstanceId()
# 实例 vpc id
request.set VpcId ("实例 VPC ID")
# 实例 ecs region eg: cn-hangzhou
request.set EcsRegion ("实例 ecs region")
# request.set Sid();
# 实例db登录用户
request.set DatabaseUser ("实例db登录用户")
# 实例db登录密码
request.set DatabasePassword ("实例db登录密码")
# 实例别名,方便DMS企业版里面区分实例用
request.set InstanceAlias ("实例别名")
# 实例dba 阿里云uid (此用户必须先在DMS企业版录入,或者使用API提前录入)
request.set DbaUid(-1)
request.set SafeRule ("DMS企业版里面配置的安全规则的规则名")
request.set QueryTimeout(60)
request.set ExportTimeout(600)
response = client.do action with exception(request)
print response
```

6.3.2. ListInstances

调用ListInstances接口检索实例列表并获取实例详情。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListInstances	系统规定参数。取值:ListInstances。
Tid	Long	否	-1	租户ID。
SearchKey	String	否	test	搜索关键字。
DbType	String	否	MySQL	数据库类型。
EnvType	String	否	product	环境类型。
InstanceSource	String	否	RDS	实例来源。
NetType	String	否	VPC	实例网络环境,取值参 考RegisterInstance接口。
InstanceState	String	否	NORMAL	实例状态,取值和解释如下: ● NORMAL: 正常使用 ● DISABLE: 禁用
PageNumber	Integer	否	1	页码。
PageSize	Integer	否	50	每页多少条数据,取值不超过100。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
InstanceList	Array of Instance		实例列表。
Instance			

数据管理 开发指南·API参考

名称	类型	示例值	描述
DataLinkName	String	dblink_test	DataLink名称。
DatabasePassw ord	String	****	数据库用户密码。
DatabaseUser	String	dbUser	数据库用户名。
Dbald	String	1	实例DBA ID。
DbaNickName	String	dbaName	DBA名称。
DdlOnline	Integer	1	实例onlineddl配置。
Ecsinstanceld	String	XXX	ECS实例ID。
EcsRegion	String	cn-hangzhou	ECS实例所属地域。
EnvType	String	product	实例环境。
ExportTimeout	Integer	600	导出超时时间。
Host	String	XXX.XXX.XXX	实例连接地址。
InstanceAlias	String	test	实例别名。
InstanceId	String	1	实例ID。
InstanceSource	String	RDS	实例来源。
InstanceType	String	mysql	实例数据库类型。
OwnerldList	List	1	数据库所属Owner ID列表。
OwnerNameList	List	11	数据库所属Owner名称列表。

名称	类型	示例值	描述
Port	Integer	3306	实例端口。
QueryTimeout	Integer	60	查询超时时间。
SafeRuleId	String	1	实例安全规则ID。
Sid	String	test	实例SID。
State	String	NORMAL	实例状态。
UseDsql	Integer	1	实例是否开启跨库查询。
Vpcld	String	XXX	VPC ID。
RequestId	String	B4B07137-F6AE- 4756-8474- 7F92BB6C4E04	请求ID。
Success	Boolean	true	请求是否成功。
TotalCount	Long	0	满足条件的实例总个数。

示例

请求示例

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

正常返回示例

XML 格式

<TotalCount>7</TotalCount>

<RequestId>98D35416-2B92-4CE5-8FD2-B1E61E165536/RequestId>

<Success>true</Success>

<InstanceList>

</InstanceList>

JSON 格式

```
"TotalCount": 7,
   "RequestId": "98D35416-2B92-4CE5-8FD2-B1E61E165536",
   "Success": true,
   "InstanceList": {
      "Instance": [
      ]
   }
}
```

6.3.3. GetInstance

调用GetInstance接口获取实例详细信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetInstance	系统规定参数。取值: GetInstance 。
Host	String	是	xxxx.mysql.rds.ali yuncs.com	实例连接地址。
Port	Integer	是	3306	实例连接端口。
Tid	Long	是	-1	租户ID。
Sid	String	否	test	实例SID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
Instance	Struct		实例详细信息。
DataLinkName	String	test	实例DataLink名称。

名称	类型	示例值	描述
Dat abasePassw ord	String	*****	实例登录账号密码。
DatabaseUser	String	dbuser	实例登录用户名。
Dbald	String	1	实例DBA ID,与ListUsers中的ID对应。
DbaNickName	String	dbaname	实例DBA名称。
DdlOnline	Integer	1	实例onlineddl配置。
EcsInstanceId	String	XXX	ECS实例ID。
EcsRegion	String	cn-hangzhou	实例所属地域。
EnvType	String	test	实例所属环境。
ExportTimeout	Integer	600	实例导出超时时间。
Host	String	xxx.rwlb.rds.aliyuncs .com	实例连接地址。
InstanceAlias	String	test	实例别名。
InstanceId	String	1	实例ID。
InstanceSource	String	RDS	实例来源。
InstanceType	String	polardb	实例DB类型。
OwnerldList	List	88656	Owner ID列表。
OwnerNameList	List	XXX	Owner名称列表。
Port	Integer	3306	实例连接端口。

名称	类型	示例值	描述
QueryTimeout	Integer	60	实例查询超时时间
SafeRuleId	String	1	实例所对应安全规则ID。
Sid	String	test	实例SID。
State	String	NORMAL	实例状态。
UseDsql	Integer	1	实例是否开启跨库查询。
Vpcld	String	xxx	实例所属VPC ID。
RequestId	String	8BE73E78-76E2- 46AA-970D- A4579A16CD4A	API请求ID。
Success	Boolean	true	接口调用是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=GetInstance
&Host=xxxx.mysql.rds.aliyuncs.com
&Port=3306
&Tid=-1

正常返回示例

&<公共请求参数>

XML 格式

```
<Instance>
   <SafeRuleId>1037</SafeRuleId>
   <UseDsql>1</UseDsql>
   <EcsInstanceId>pc-xxx</EcsInstanceId>
   <QueryTimeout>80</QueryTimeout>
   <Host>xxx.rwlb.rds.aliyuncs.com
   <InstanceAlias>test</InstanceAlias>
   <InstanceSource>RDS</InstanceSource>
   <InstanceId>24710</InstanceId>
   <DataLinkName>dblink_test
   <EnvType>dev</EnvType>
   <DdlOnline>0</DdlOnline>
   <VpcId>vpc-xxx</VpcId>
   <InstanceType>polardb</InstanceType>
   <DbaId>104442
   <Port>3306</Port>
   <DbaNickName>test[子账号]
   <State>NORMAL</State>
   <DatabaseUser>dmstest/DatabaseUser>
   <DatabasePassword>*****
   <EcsRegion>cn-hangzhou</EcsRegion>
   <ExportTimeout>300</ExportTimeout>
</Instance>
<RequestId>8BE73E78-76E2-46AA-970D-A4579A16CD4A/RequestId>
<Success>true</Success>
```

JSON 格式

数据管理 开发指南·API参考

```
"Instance": {
 "SafeRuleId": 1037,
 "UseDsql": "1",
  "EcsInstanceId": "pc-xxx",
  "QueryTimeout": 80,
  "Host": "xxx.rwlb.rds.aliyuncs.com",
 "InstanceAlias": "test",
  "InstanceSource": "RDS",
 "InstanceId": 24710,
 "DataLinkName": "dblink test",
  "EnvType": "dev",
  "DdlOnline": 0,
  "VpcId": "vpc-xxx",
  "InstanceType": "polardb",
 "DbaId": 104442,
  "Port": 3306,
 "DbaNickName": "test[子账号]",
 "State": "NORMAL",
  "DatabaseUser": "dmstest",
  "DatabasePassword": "*****",
 "EcsRegion": "cn-hangzhou",
 "ExportTimeout": 300
"RequestId": "8BE73E78-76E2-46AA-970D-A4579A16CD4A",
"Success": true
```

6.3.4. UpdateInstance

调用UpdateInstance接口更新实例信息,同时会检查连通性。

调用UpdateInstance更新实例,请先通过GetInstance API获取对应实例的完整信息,再根据具体需要修改的内容进行设置,然后再调用此API

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateInstance	系统规定参数。取 值:UpdateInstance。
DatabasePasswor d	String	是	*****	数据库账号密码。
DatabaseUser	String	是	dbuser	数据库账号。
Dbald	String	是	1	实例DBA ID,与ListUsers接口中的ID对应。

名称	类型	是否必选	示例值	描述
EnvType	String	是	test	实例所属环境。
ExportTimeout	Integer	是	600	实例导出超时时间。
Host	String	是	xxxx.mysql.rds.ali yuncs.com	实例连接地址。
InstanceAlias	String	是	test	实例别名。
InstanceId	String	是	1	实例ID,与 GetInstance 中返回的ID对 应。
InstanceSource	String	是	RDS	实例来源,取值可参 考RegisterInstance中的提示。
InstanceType	String	是	MySQL	实例数据库类型,取值可参 考RegisterInstance中的提示。
Port	Integer	是	3306	实例端口。
QueryTimeout	Integer	是	60	实例查询超时时间。
SafeRuleId	String	是	1	实例所属安全规则ID。
Tid	Long	是	-1	租户ID。
EcsInstanceId	String	否	xxx	ECS实例ID。
Vpcld	String	否	xxx	实例所属VPC ID。
EcsRegion	String	否	cn-hangzhou	实例所属地域。
Sid	String	否	xxx	实例SID。
Dat aLinkName	String	否	datalink_test	实例DataLink名称。

名称	类型	是否必选	示例值	描述
DdlOnline	Integer	否	1	实例onlineddl设置,取值请参 考RegisterInstance接口。
UseDsql	Integer	否	1	实例是否启用跨库查询。
SkipTest	Boolean	否	true	是否跳过测试连接,取值如下: • true: 跳过测试连接。 • false: 不跳过测试连接。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	E9BEBF41-4F69- 4605-A5D5- A67955173941	API请求ID。
Success	Boolean	false	更新实例是否成功。

示例

请求示例

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

正常返回示例

XML 格式

<RequestId>E9BEBF41-4F69-4605-A5D5-A67955173941</RequestId><Success>true</Success>

JSON 格式

```
{
   "RequestId": "E9BEBF41-4F69-4605-A5D5-A67955173941",
   "Success": true
}
```

6.3.5. DeleteInstance

调用DeleteInstance接口删除DMS企业版管理的实例。

注意此删除是指将DMS企业版中录入的实例从企业版的管理里面移除,不会导致您的实例下线

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteInstance	系统规定参数,取 值:DeleteInstance。
Host	String	是	xxx.mysql.rds.aliy uncs.com	待删除的实例连接地址。
Port	Integer	是	3306	待删除的实例链接端口。
Tid	Long	是	-1	租户ID。
Sid	String	否	testSid	待删除的实例SID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误消息。
RequestId	String	B4B07137-F6AE- 4756-8474- 7F92BB6C4E04	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=DeleteInstance &Host=xxx.mysql.rds.aliyuncs.com &Port=3306 &Tid=-1 &<公共请求参数>

正常返回示例

XML 格式

<RequestId>B4B07137-F6AE-4756-8474-7F92BB6C4E04</RequestId> <ErrorCode>UnknownError</ErrorCode>

<ErrorMessage>UnknownError

<Success>true</Success>

JSON 格式

{"RequestId": "B4B07137-F6AE-4756-8474-7F92BB6C4E04", "ErrorCode": "UnknownError", "ErrorMessage": "UnknownError", "Success": "true"}

6.4. 操作日志

6.4.1. GetOpLog

调用GetOpLog接口获取指定时间内的操作日志详情列表。

账户角色为管理员、安全管理员可调用该接口,关于账号角色的更多信息,请参见用户管理。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetOpLog	系统规定参数。取值: Get OpLog 。
Tid	Long	是	-1	租户ID。
StartTime	String	是	2018-11-01 10:00:00	开始时间,时间格式为*yyyy-MM-DD HH:mm:ss*。
EndTime	String	是	2018-11-01 11:00:00	结束时间,时间格式为*yyyy-MM-DD HH:mm:ss*。
PageNumber	Integer	是	3	页码,从1开始。

名称	类型	是否必选	示例值	描述
PageSize	Integer	是	30	每页记录数,取值: • 30 • 50 • 100
Module	String	否	PERMISSION	日志模块,缺省则对应时间段内所有模块返回,日志模块取值和说明如下: PERMISSION: 权限 OWNER: 数据Owner SQL_CONSOLE: 数据查询 SQL_CONSOLE_EXPORT: 查询结果导出 DATA_CHANGE: 数据变更 DATA_EXPORT: 数据导出 SQL_REVIEW: SQL审核 DT_SYNC: 库表同步 DT_DETAIL: 库表详情 DB_TASK: 任务管理 INSTANCE_MANAGE: 实例管理 SECURITY_RULE: 安全规则 CONFIG_MANAGE: 配置管理 RESOURCE_AUTH: 资源授权 ACCESS_WHITE_IP: 访问IP白名单
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	403	错误码。
ErrorMessage	String	请求参数不合法	错误消息。
OpLogDetails	Array of OpLogDetail		操作日志详情列表。
OpLogDetail			

数据管理 开发指南·API参考

名称	类型	示例值	描述
Database	String	dmstest_prod_data base@dmstest.rds .【测试实例】	数据库访问地址,只有当实例类型为LocalInstance时,这个参数才有值。 ② 说明 仅任务类型区域的各功能模块操作才有这个属性。
Module	String	PERMISSION	功能模块。
OpContent	String	申请权限	操作内容。
OpTime	String	2018-11-01 10:00:00	操作时间。
Orderld	Long	1	工单号或任务号。 ② 说明 仅任务类型区域的各功能模块操作与系统管理任务管理功能模块的操作才有这个属性。
Userld	Long	12345678	阿里云UID。
UserNick	String	dmstest	用户显示名。
RequestId	String	47D56208-DB1D- 4FD3-BE32- 300E43185488	请求唯一ID。
Success	Boolean	true	请求是否成功。
TotalCount	Long	22	操作日志总条数。

示例

请求示例

http(s)://[Endpoint]/?Action=GetOpLog &Tid=-1 &StartTime=2018-11-01 10:00:00 &EndTime=2018-11-01 11:00:00 &PageNumber=3 &PageSize=30 &<公共请求参数>

开发指南·API参考 数据管理

正常返回示例

XML 格式

```
<OpLogDetails>
   <OpLogDetail>
       <OpContent>(成功,行数:1)SELECT (data length + index length) AS store_capacity FROM i
nformation schema.tables WHERE table type = & #39; BASE TABLE& #39; AND `table schema`
= & #39; dmstest_database& #39; AND `table_name` = & #39; multi_db_multi_tbl& #39
;</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 20:04:10</opTime>
       <Module>数据查询</Module>
   </OpLogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:1)SELECT (data length + index length) AS store capacity FROM i
nformation schema.tables WHERE table type = & #39; BASE TABLE& #39; AND `table schema`
= & #39; dmstest_database& #39; AND `table_name` = & #39; multi_db_multi_tbl& #39
;</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 20:04:06</pTime>
       <Module>数据查询</Module>
   </OplogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:24)show topology FROM multi db multi tbl</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 20:03:55
       <Module>数据查询</Module>
   </OplogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:1) SELECT (data length + index length) AS store capacity FROM i
nformation schema.tables WHERE table type = & #39; BASE TABLE& #39; AND `table schema`
= & #39; dmstest database& #39; AND `table name` = & #39; user log3& #39; </op>
ent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest_database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 20:02:59</opTime>
       <Module>数据查询</Module>
   </OplogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:1) SELECT (data_length + index_length) AS store_capacity FROM i
nformation schema.tables WHERE table type = & #39; BASE TABLE& #39; AND `table schema`
= & #39; dmstest database& #39; AND `table name` = & #39; user log3& #39; </op>
ent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 20:02:49</opTime>
```

```
<Module>数据查询</Module>
   </OplogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:1) show CREATE TABLE user log3</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest_database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:34:05</pTime>
       <Module>数据查询</Module>
   </OpLogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:9)SELECT `TABLE NAME` FROM information schema.tables WHERE TAB
LE TYPE = & #39; BASE TABLE& #39; AND table schema = & #39; dmstest database& #39
;</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:33:54
       <Module>数据查询</Module>
   </OplogDetail>
   <OpLogDetail>
       <OpContent>新增SQL任务,任务号: 5486</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:26:57</opTime>
       <Module>任务管理</Module>
   </OpLogDetail>
   <OpLogDetail>
       <OpContent>新增SQL任务,任务号: 5485</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:23:55
       <Module>任务管理</Module>
   </OplogDetail>
   <OpLogDetail>
       <OpContent>新增SQL任务,任务号: 5484</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:23:30</pTime>
       <Module>任务管理</Module>
   </OpLogDetail>
   <OpLogDetail>
       <OpContent>新增SQL任务,任务号: 5483</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:23:11
       <Module>任务管理</Module>
   </OplogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:1) show CREATE TABLE multi db single tbl</OpContent>
```

```
<UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:22:30</opTime>
       <Module>数据查询</Module>
   </OpLogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:2)SELECT `TABLE NAME` FROM information schema.tables WHERE TAB
LE TYPE = & #39; BASE TABLE& #39; AND table schema = & #39; dmstest database& #39
;</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:22:15</pTime>
       <Module>数据查询</Module>
   </OplogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:2)SELECT `TABLE NAME` FROM information schema.tables WHERE TAB
LE TYPE = & #39; BASE TABLE& #39; AND table schema = & #39; dmstest database& #39
;</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest_database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:19:31</pTime>
       <Module>数据查询</Module>
   </OpLogDetail>
   <OpLogDetail>
       <OpContent>新增SQL任务,任务号: 5482</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:19:25
       <Module>任务管理</Module>
   </OpLogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:1)SELECT `TABLE_NAME` FROM information_schema.tables WHERE TAB
LE_TYPE = & #39; BASE TABLE& #39; AND table_schema = & #39; dmstest_database& #39
:</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:18:57</opTime>
       <Module>数据查询</Module>
   </OpLogDetail>
   <OpLogDetail>
       <OpContent>新增SQL任务,任务号: 5481</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:18:46</pTime>
       <Module>任务管理</Module>
   </OplogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:0)SELECT `TABLE NAME` FROM information schema.tables WHERE TAB
```

```
LE TYPE = & #39; BASE TABLE& #39; AND table schema = & #39; dmstest database& #39
;</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:04:13</pTime>
       <Module>数据查询</Module>
   </OplogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:0)SELECT `TABLE NAME` FROM information schema.tables WHERE TAB
LE TYPE = & #39; BASE TABLE& #39; AND table schema = & #39; dmstest database& #39
;</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 14:04:08</pTime>
       <Module>数据查询</Module>
   </OpLogDetail>
   <OpLogDetail>
       <OpContent>(成功,行数:0)SELECT `TABLE NAME` FROM information schema.tables WHERE TAB
LE TYPE = & #39; BASE TABLE& #39; AND table schema = & #39; dmstest database& #39
;</OpContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 12:11:41</pTime>
       <Module>数据查询</Module>
   </OplogDetail>
   <OpLogDetail>
       <opContent>管理员或DBA编辑库OWNER, 库名: dmstest database@dmstest.rds...(test)
ent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest database@dmstest.rds...(test)/Database>
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 12:11:14</pTime>
       <Module>数据Owner</Module>
   </OpLogDetail>
   <OpLogDetail>
       <opContent>管理员或DBA编辑库OWNER, 库名: dmstest prod database@dmstest.rds...(test)/o
pContent>
       <UserNick>dmstest</UserNick>
       <Database>dmstest_prod_database@dmstest.rds...(test)
       <UserId>12345678</UserId>
       <OpTime>2018-11-22 12:11:07</opTime>
       <Module>数据Owner</Module>
   </OplogDetail>
</OpLogDetails>
<TotalCount>22</TotalCount>
<RequestId>47D56208-DB1D-4FD3-BE32-300E43185488/RequestId>
<Success>true</Success>
```

```
JSON 格式
```

```
"OpLogDetails": {
  "OpLogDetail": [{
   "OpContent": "(成功,行数:1)SELECT (data length + index length) AS store capacity FROM inf
ormation_schema.tables WHERE table_type = 'BASE TABLE' AND `table schema` = 'dm
stest database' AND `table name` = ' multi db multi tbl'",
   "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
   "UserId": 12345678,
   "OpTime": "2018-11-22 20:04:10",
   "Module": "数据查询"
   "OpContent": "(成功,行数:1)SELECT (data_length + index_length) AS store_capacity FROM inf
ormation schema.tables WHERE table type = 'BASE TABLE' AND `table schema` = 'dm
stest database' AND `table name` = ' multi db multi tbl'",
   "UserNick": "dmstest",
   "Database": "dmstest_database@dmstest.rds...(test)",
   "UserId": 12345678,
   "OpTime": "2018-11-22 20:04:06",
  "Module": "数据查询"
   "OpContent": "(成功,行数:24) show topology FROM multi db multi tbl",
   "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
   "UserId": 12345678,
   "OpTime": "2018-11-22 20:03:55",
  "Module": "数据查询"
   "OpContent": "(成功,行数:1)SELECT (data length + index length) AS store capacity FROM inf
ormation_schema.tables WHERE table_type = 'BASE TABLE' AND `table schema` = 'dm
stest database' AND `table name` = 'user log3'",
   "UserNick": "dmstest",
   "Database": "dmstest database@dmstest.rds...(test)",
   "UserId": 12345678,
   "OpTime": "2018-11-22 20:02:59",
  "Module": "数据查询"
   "OpContent": "(成功,行数:1)SELECT (data length + index length) AS store capacity FROM inf
ormation schema.tables WHERE table type = 'BASE TABLE' AND `table schema` = 'dm
stest database' AND `table name` = 'user log3'",
   "UserNick": "dmstest",
   "Database": "dmstest database@dmstest.rds...(test)",
   "UserId": 12345678,
  "OpTime": "2018-11-22 20:02:49",
  "Module": "数据查询"
   "OpContent": "(成功,行数:1)show CREATE TABLE user_log3",
   "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
   "UserId": 12345678,
   "OpTime": "2018-11-22 14:34:05",
  "Module": "数据查询"
   "OpContent": "(成功,行数:9)SELECT `TABLE NAME` FROM information schema.tables WHERE TABLE
TYPE = 'BASE TABLE' AND table schema = 'dmstest database'",
```

```
"UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:33:54",
  "Module": "数据查询"
  "OpContent": "新增SQL任务,任务号: 5486",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:26:57",
  "Module": "任务管理"
  "OpContent": "新增SQL任务,任务号: 5485",
  "UserNick": "dmstest",
  "Database": "dmstest_database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:23:55",
  "Module": "任务管理"
  "OpContent": "新增SQL任务,任务号: 5484",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:23:30",
  "Module": "任务管理"
  "OpContent": "新增SQL任务,任务号: 5483",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:23:11",
  "Module": "任务管理"
  "OpContent": "(成功,行数:1) show CREATE TABLE multi db single tbl",
  "UserNick": "dmstest",
  "Database": "dmstest_database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:22:30",
  "Module": "数据查询"
 }, {
  "OpContent": "(成功,行数:2)SELECT `TABLE NAME` FROM information schema.tables WHERE TABLE
_TYPE = ' BASE TABLE' AND table_schema = ' dmstest_database'",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:22:15",
  "Module": "数据查询"
  "OpContent": "(成功,行数:2)SELECT `TABLE NAME` FROM information schema.tables WHERE TABLE
TYPE = 'BASE TABLE' AND table schema = 'dmstest database'",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
```

```
"OpTime": "2018-11-22 14:19:31",
  "Module": "数据查询"
 }, {
  "OpContent": "新增SQL任务,任务号: 5482",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:19:25",
  "Module": "任务管理"
  "OpContent": "(成功,行数:1)SELECT `TABLE NAME` FROM information schema.tables WHERE TABLE
TYPE = 'BASE TABLE' AND table schema = 'dmstest database'",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:18:57",
  "Module": "数据查询"
 }, {
  "OpContent": "新增SQL任务,任务号: 5481",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:18:46",
  "Module": "任务管理"
  "OpContent": "(成功,行数:0)SELECT `TABLE NAME` FROM information schema.tables WHERE TABLE
TYPE = ' BASE TABLE' AND table schema = ' dmstest database'",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:04:13",
  "Module": "数据查询"
  "OpContent": "(成功,行数:0)SELECT `TABLE NAME` FROM information schema.tables WHERE TABLE
TYPE = 'BASE TABLE' AND table schema = 'dmstest database'",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 14:04:08",
  "Module": "数据查询"
  "OpContent": "(成功,行数:0)SELECT `TABLE NAME` FROM information schema.tables WHERE TABLE
TYPE = 'BASE TABLE' AND table schema = 'dmstest database'",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 12:11:41",
  "Module": "数据查询"
  "OpContent": "管理员或DBA编辑库OWNER, 库名: dmstest database@dmstest.rds...(test)",
  "UserNick": "dmstest",
  "Database": "dmstest database@dmstest.rds...(test)",
  "UserId": 12345678,
  "OpTime": "2018-11-22 12:11:14",
```

```
"Module": "致据Owner"
}, {
   "OpContent": "管理员或DBA编辑库OWNER, 库名: dmstest_prod_database@dmstest.rds...(test)",
   "UserNick": "dmstest",
   "Database": "dmstest_prod_database@dmstest.rds...(test)",
   "UserId": 12345678,
   "OpTime": "2018-11-22 12:11:07",
   "Module": "数据Owner"
}]
},
   "TotalCount": 22,
   "RequestId": "47D56208-DB1D-4FD3-BE32-300E43185488",
   "Success": true
}
```

SDK示例

SDK使用基础信息参考: 阿里云Python SDK

示例如下:

6.5. 审批流程

6.5.1. ListWorkFlowTemplates

调用ListWorkFlowTemplates接口获取审批模板列表。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListWorkFlowTe mplates	系统规定参数。取 值:ListWorkFlowTemplates。

名称	类型	是否必选	示例值	描述
Tid	Long	是	-1	租户ID。
SearchName	String	否	Admin	搜索模板名。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误消息。
RequestId	String	41067071-0243- 4AAB-B3CF- 4DE6D54F53B4	请求ID。
Success	Boolean	true	请求是否成功。
WorkFlowTemp lates	Array of WorkFlowTemp late		审批模板列表。
WorkFlowTemp late			
Comment	String	test	备注。
CreateUserld	Long	1234	创建人ID。
Enabled	String	Υ	模板是否启用,取值和说明如下: Y: 启用 N: 禁用
lsSystem	Integer	1	是否系统定义的审批模板,取值和说明如下: 1: 是 0: 否
TemplateId	Long	12345	模板ID。

名称	类型	示例值	描述
TemplateName	String	Admin	模板名称。
WorkflowNode s	Array of WorkflowNode		审批节点信息。
WorkflowNode			
Comment	String	test	备注。
CreateUserId	Long	1234	创建人ID。
Nodeld	Long	123	节点ID。
NodeName	String	test	节点名称。
NodeType	String	SYS	节点类型,取值和说明如下: ● SYS: 系统定义 ● USER_LIST: 用户自定义
Position	Integer	1	顺序。
TemplateId	Long	12345	模板ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ListWorkFlowTemplates &Tid=-1 &<公共请求参数>
```

正常返回示例

XML 格式

```
<Position>0</Position>
               <NodeType>SYS</NodeType>
               <NodeId>3763</NodeId>
               <TemplateId>6243</TemplateId>
           </WorkFlowNode>
       </WorkFlowNodes>
       <TemplateName>Admin</TemplateName>
       <Enabled>Y</Enabled>
       <IsSystem>1</IsSystem>
       <TemplateId>6243</TemplateId>
   </WorkFlowTemplate>
   <WorkFlowTemplate>
       <Comment>系统定义审批线,先数据Owner审批,再DBA审批,最后Admin审批</Comment>
       <WorkflowNodes>
           <WorkFlowNode>
               <Comment>系统节点,动态计算,由用户操作的资源对应的Owner来完成审批</Comment>
               <NodeName>Owner</NodeName>
               <Position>0</Position>
               <NodeType>SYS</NodeType>
               <NodeId>3762</NodeId>
               <TemplateId>6244</TemplateId>
           </WorkFlowNode>
           <WorkFlowNode>
               <Comment>系统节点,动态计算,由用户操作的资源对应的DBA来完成审批</Comment>
               <NodeName>DBA</NodeName>
               <Position>0</Position>
               <NodeType>SYS</NodeType>
               <NodeId>3761</NodeId>
               <TemplateId>6244</TemplateId>
           </WorkFlowNode>
           <WorkFlowNode>
               <Comment>系统节点,动态计算,由管理员来完成审批</Comment>
               <NodeName>Admin</NodeName>
               <Position>0</Position>
               <NodeType>SYS</NodeType>
               <NodeId>3763</NodeId>
               <TemplateId>6244</TemplateId>
           </WorkFlowNode>
       </WorkflowNodes>
       <TemplateName>Owner--&gt;DBA--&gt;Admin</TemplateName>
       <Enabled>Y</Enabled>
       <IsSystem>1</IsSystem>
       <TemplateId>6244</TemplateId>
   </WorkFlowTemplate>
</WorkFlowTemplates>
```

JSON 格式

```
"Comment": "系统定义审批线,仅需管理员审批",
  "WorkFlowNodes": {
   "WorkFlowNode": [
       "Comment": "系统节点,动态计算,由管理员来完成审批",
       "NodeName": "Admin",
       "Position": 0,
       "NodeType": "SYS",
       "NodeId": 3763,
       "TemplateId": 6243
   ]
 },
 "TemplateName": "Admin",
 "Enabled": "Y",
 "IsSystem": 1,
 "TemplateId": 6243
},
 "Comment": "系统定义审批线,先数据Owner审批,再DBA审批,最后Admin审批",
 "WorkflowNodes": {
   "WorkFlowNode": [
     {
       "Comment": "系统节点,动态计算,由用户操作的资源对应的Owner来完成审批",
       "NodeName": "Owner",
       "Position": 0,
       "NodeType": "SYS",
       "NodeId": 3762,
       "TemplateId": 6244
     },
       "Comment": "系统节点,动态计算,由用户操作的资源对应的DBA来完成审批",
       "NodeName": "DBA",
       "Position": 0,
       "NodeType": "SYS",
       "NodeId": 3761,
       "TemplateId": 6244
     },
       "Comment": "系统节点,动态计算,由管理员来完成审批",
       "NodeName": "Admin",
       "Position": 0,
       "NodeType": "SYS",
       "NodeId": 3763,
       "TemplateId": 6244
   ]
  },
 "TemplateName": "Owner-->DBA-->Admin",
 "Enabled": "Y",
 "IsSystem": 1,
 "TemplateId": 6244
```

```
}
```

6.5.2. ListWorkFlowNodes

调用ListWorkFlowNodes接口获取审批节点列表。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListWorkFlowNod es	系统规定参数。取 值:ListWorkFlowNodes。
Tid	Long	是	-1	租户ID。
SearchName	String	否	admin	搜索审批节点名称。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误消息。
RequestId	String	CFD8FE00-36D9- 4C1B-940D- 65A7B73D9066	请求ID。
Success	Boolean	true	请求是否成功。
WorkflowNode s	Array of WorkflowNode		审批节点信息。
WorkflowNode			
AuditUsers	Array of AuditUser		审批人信息。
AuditUser			

数据管理 开发指南·API参考

名称	类型	示例值	描述
NickName	String	test	审批人名称。
RealName	String	test	审批人真实名。
Userld	Long	123	审批人用户ID,注意不是阿里云UID。
Comment	String	test	备注。
CreateUserld	Long	123	创建者用户ID,注意不是阿里云UID。
CreateUserNick Name	String	test	创建人名称。
Nodeld	Long	123	节点ID。
NodeName	String	test	节点名称。
NodeType	String	SYS	节点类型,取值和说明如下: ● SYS: 系统定义 ● USER_LIST: 用户自定义

示例

请求示例

http(s)://[Endpoint]/?Action=ListWorkFlowNodes
&Tid=-1

&<公共请求参数>

正常返回示例

XML 格式

JSON 格式

6.6. 库表元数据

6.6.1. SearchDatabase

调用SearchDatabase接口搜索库详情信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	SearchDatabase	系统规定参数。取 值:SearchDatabase。
PageNumber	Integer	是	1	页码。

名称	类型	是否必选	示例值	描述
SearchKey	String	是	testdb	搜索数据库名称的关键字。
Tid	Long	是	1	租户ID。
PageSize	Integer	否	10	每页记录数。
EnvType	String	否	product	数据库的环境类型。
SearchRange	String	否	HAS_PERMSSION	权限范围,取值和说明: HAS_PERMSSION:有权限的 OWNER:我所有的 MY_FOCUS:我关注的 UNKNOWN:不限
SearchTarget	String	否	SINGLE_DB	搜索目标,取值和说明如下: DB: 不限(不区分单库或逻辑库) SINGLE_DB: 单库 LOGIC_DB: 逻辑库
DbType	String	否	MySQL	数据库类型,取值为: MySQL SQLServer PostgreSQL Oracle DRDS OceanBase Mongo Redis
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。

名称	类型	示例值	描述
RequestId	String	E0D21075-CD3E- 4D98-8264- FD8AD04A63B6	API请求ID。
SearchDat abas eList	Array of SearchDatabas e		搜索库详情列表。
SearchDatabas e			
Alias	String	test	数据库别名。
Databaseld	String	1	数据库ID。
DatalinkName	String	datalink_name	数据库所对应DataLink。
DbType	String	mysql	数据库类型,取值为: MySQL SQLServer PostgreSQL Oracle DRDS OceanBase Mongo Redis
Dbald	String	1	数据库的DBA ID。
Encoding	String	utf8	数据库编码。
EnvType	String	product	数据库的环境类型。
Host	String	xxx.xxx.xxx	库所在的实例对应连接地址。
Logic	Boolean	false	是否为逻辑库,取值: • true: 是 • false: 否
OwnerldList	List	1	数据库Owner ID列表。

名称	类型	示例值	描述
OwnerNameList	List	user1	数据库Owner名称列表。
Port	Integer	3306	数据库所对应实例端口。
SchemaName	String	test	数据库名。
SearchName	String	test@xxx.xxx.xxx :3306	数据库搜索名。
Sid	String	testSid	库所对应实例SID。
TotalCount	Long	1	满足条件的记录总数。
Success	Boolean	true	请求是否成功。

示例

请求示例

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

&PageNumber=1

&SearchKey=testdb

&Tid=-1

&<公共请求参数>

正常返回示例

XML 格式

```
<TotalCount>1</TotalCount>
<RequestId>E0D21075-CD3E-4D98-8264-FD8AD04A63B6/RequestId>
<SearchDatabaseList>
   <SearchDatabase>
       <Encoding>utf8mb4</Encoding>
       <Host>pc-xxx.mysql.polardb.rds.aliyuncs.com</Host>
       <DbType>polardb</DbType>
       <EnvType>dev</EnvType>
       <OwnerIdList>
           <OwnerIds>88653</OwnerIds>
           <OwnerIds>88656</OwnerIds>
           <OwnerIds>104442</OwnerIds>
       </OwnerIdList>
       <Logic>false</Logic>
       <DbaId>88653/DbaId>
       <OwnerNameList>
           <OwnerNames>dmstest</OwnerNames>
           <OwnerNames>xx</OwnerNames>
           <OwnerNames>xx[子账号]</OwnerNames>
       </OwnerNameList>
       <Port>3306</Port>
       <DatabaseId>117567</DatabaseId>
       <Alias>xx线下库</Alias>
       <SearchName>xxx@pc-xxx.mysql.polardb.rds.aliyuncs.com:3306【xx线下库】</SearchName>
       <DatalinkName>dblink 38
       <SchemaName>xxx_0000</SchemaName>
   </SearchDatabase>
</SearchDatabaseList>
<Success>true</Success>
```

JSON 格式

```
"TotalCount": 1,
"RequestId": "E0D21075-CD3E-4D98-8264-FD8AD04A63B6",
"SearchDatabaseList": {
  "SearchDatabase": [
      "Encoding": "utf8mb4",
      "Host": "pc-xxx.mysql.polardb.rds.aliyuncs.com",
      "DbType": "polardb",
      "EnvType": "dev",
      "OwnerIdList": {
        "OwnerIds": [
          88653,
         88656,
         104442
       ]
      "Logic": false,
      "DbaId": 88653,
      "OwnerNameList": {
        "OwnerNames": [
          "dmstest",
         "xx",
         "xx[子账号]"
      },
      "Port": 3306,
      "DatabaseId": 117567,
      "Alias": "xx线下库",
      "SearchName": "xxx@pc-xxx.mysql.polardb.rds.aliyuncs.com:3306【xx线下库】",
      "DatalinkName": "dblink 38",
      "SchemaName": "xxx_0000"
    }
 ]
},
"Success": true
```

6.6.2. SearchTable

调用SearchTable接口搜索表详情列表。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	SearchT able	系统规定参数。取值:SearchTable。
Tid	Long	否	1	租户ID。

名称	类型	是否必选	示例值	描述
SearchKey	String	否	test	搜索表名称的关键字。
PageNumber	Integer	否	1	第几页。
PageSize	Integer	否	10	每页记录数。
EnvType	String	否	PRODUCT	数据库的环境类型。
SearchTarget	String	否	LOGIC_TABLE	搜索表类型,取值如下: TABLE:表,包含物理表和逻辑表 SINGLE_TABLE:物理表 LOGIC_TABLE:逻辑表
SearchRange	String	否	OWNER	权限范围,取值和说明: HAS_PERMSSION:有权限的 OWNER:我所有的 MY_FOCUS:我关注的 UNKNOWN:不限
DbType	String	否	MySQL	数据库类型,取值为: MySQL SQLServer PostgreSQL Oracle DRDS OceanBase Mongo Redis
ReturnGuid	Boolean	否	false	是否返回表GUID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。

名称 类型	示例值	描述
-------	-----	----

ErrorMessage	String	UnknownError	错误信息。
RequestId	String	1489257F-1B5D- 4B5B-89EF- 923C12CEEBD1	请求ID。
SearchT ableList	Array of SearchTable		搜索表详情列表。
SearchTable			
DBSearchName	String	test	表所对应的库搜索名。
Databaseld	String	1	所属的库ID。
DbName	String	test	数据库名。
DbType	String	mysql	数据库类型,取值为: • MySQL • SQLServer • PostgreSQL • Oracle • DRDS • OceanBase • Mongo • Redis
Description	String	test	描述信息。
Encoding	String	utf8	表编码。
Engine	String	innodb	表引擎。
EnvType	String	test	环境类型。

名称	类型	示例值	描述
Logic	Boolean	false	是否为逻辑表,取值: • true: 是 • false: 否
OwnerldList	List	1	表Owner ID列表。
OwnerNameList	List	user	表Owner昵称列表。
TableGuid	String	IDB_L_9032.db- test.yuyang_test	表GUID。
Tableid	String	1	表ID。
TableName	String	test_table	表名。
TableSchemaN ame	String	test@xxx.xxx.xxx :3306	表所对应库名。
Success	Boolean	true	请求是否成功。
TotalCount	Long	1	满足搜索条件的记录总数。

示例

请求示例

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

正常返回示例

XML 格式

数据管理 开发指南·API参考

```
<SearchTableList>
   <SearchTable>
       <TableGuid>IDB_L_9032.db-test.yuyang_test</TableGuid>
       <Logic>false</Logic>
       <TableSchemaName>xxx 0000</TableSchemaName>
       <OwnerNameList>
           <OwnerNames>xxx</OwnerNames>
            <OwnerNames>xxx</OwnerNames>
            <OwnerNames>xxx</OwnerNames>
       </OwnerNameList>
       <DBSearchName>xxx 0000@pc-xxx.mysql.polardb.rds.aliyuncs.com:3306【xxx线下库】
archName>
       <DatabaseId>117567/DatabaseId>
       <Encoding>utf8mb4</Encoding>
       <TableName>test_aaaa</TableName>
       <OwnerIdList>
           <OwnerIds>88653</OwnerIds>
           <OwnerIds>88656</OwnerIds>
           <OwnerIds>104442</OwnerIds>
       </OwnerIdList>
       <EnvType>dev</EnvType>
       <Engine>InnoDB</Engine>
       <TableId>18396767</TableId>
   </SearchTable>
</SearchTableList>
<TotalCount>1</TotalCount>
<RequestId>1489257F-1B5D-4B5B-89EF-923C12CEEBD1</RequestId>
<Success>true</Success>
```

JSON 格式

```
"SearchTableList": {
  "SearchTable": [
      "TableGuid": "IDB L 9032.db-test.yuyang test",
      "Logic": false,
      "TableSchemaName": "xxx 0000",
      "OwnerNameList": {
       "OwnerNames": [
         "xxx",
         "xxx",
         "xxx"
      },
      "DBSearchName": "xxx 0000@pc-xxx.mysql.polardb.rds.aliyuncs.com:3306 [xxx线下库] ",
      "DatabaseId": 117567,
      "Encoding": "utf8mb4",
      "TableName": "test aaaa",
      "OwnerIdList": {
       "OwnerIds": [
         88653,
         88656,
        104442
       ]
      },
      "EnvType": "dev",
     "Engine": "InnoDB",
      "TableId": 18396767
 ]
},
"TotalCount": 1,
"RequestId": "1489257F-1B5D-4B5B-89EF-923C12CEEBD1",
"Success": true
```

6.6.3. ListDatabases

调用List Dat abases接口获取指定实例下的数据库列表详情信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	List Dat abases	系统规定参数。取值:ListDatabases。
InstanceId	String	是	1	实例ID,取值来自ListInstances接口返回的实例信息中的实例ID,注意不是RDS的实例ID。

名称	类型	是否必选	示例值	描述
PageNumber	Integer	是	1	页码。
Tid	Long	是	-1	租户ID。
PageSize	Integer	否	10	每页记录数。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
DatabaseList	Array of Database		数据库详情列表。
Database			
CatalogName	String	1	数据库目录名称。 ② 说明 若数据库类型为PG系列,此处
			表示PG数据库名。
Databaseld	String	1	数据库ID。
DbType	String	mysql	数据库类型。
Dbald	String	1	数据库所属DBA ID。
DbaName	String	dba_user	数据库所属DBA名称。
Encoding	String	utf-8	数据库编码。
EnvType	String	dev	数据库所属环境类型。
Host	String	XXX.XXX.XXX	实例连接地址。
InstanceId	String	1	数据库所属实例ID。

名称	类型	示例值	描述
OwnerldList	List	1	数据库Owner ID列表。
OwnerNameList	List	owner_name	数据库Owner名称列表。
Port	Integer	3306	实例连接端口。
SchemaName	String	test	数据库名。
SearchName	String	test@xxx.xxx.xxx :3306	数据库搜索名。
Sid	String	test	库所属实例SID。
State	String	NORMAL	库状态,取值和说明如下: NORMAL: 正常 DISABLE: 禁用 OFFLINE: 下线 NOT_EXIST: 不存在
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	5541CDA6-F674- 435C-81BD- 40C2FB926CE3	请求ID。
Success	Boolean	true	请求是否成功。
TotalCount	Long	1	实例下面的数据库个数。

示例

请求示例

http(s)://[Endpoint]/?Action=ListDatabases
&InstanceId=1

&PageNumber=1

&Tid=-1

&<公共请求参数>

数据管理 开发指南·API参考

正常返回示例

XML 格式

```
<TotalCount>1</TotalCount>
<RequestId>5541CDA6-F674-435C-81BD-40C2FB926CE3/RequestId>
<Success>true</Success>
<DatabaseList>
   <Database>
       <Encoding>utf8</Encoding>
       <Host>pc-xxx.rwlb.rds.aliyuncs.com</Host>
       <InstanceId>24710</InstanceId>
       <DbType>polardb</pbType>
       <EnvType>product</EnvType>
       <OwnerIdList>
           <OwnerIds>88653</OwnerIds>
       </OwnerIdList>
       <DbaId>104442</pbaId>
       <OwnerNameList>
           <OwnerNames>dmstest</OwnerNames>
       </OwnerNameList>
       <Port>3306</Port>
       <DatabaseId>116366/DatabaseId>
       <State>NORMAL</State>
       <SearchName>xxx_test@pc-xxx.rwlb.rds.aliyuncs.com:3306 [xxx_test] 
       <DbaName>xxx[子账号]</DbaName>
       <SchemaName>xxx test</SchemaName>
    </Database>
</DatabaseList>
```

JSON 格式

```
"TotalCount": 1,
"RequestId": "5541CDA6-F674-435C-81BD-40C2FB926CE3",
"Success": true,
"DatabaseList": {
  "Database": [
      "Encoding": "utf8",
      "Host": "pc-xxx.rwlb.rds.aliyuncs.com",
      "InstanceId": 24710,
      "DbType": "polardb",
      "EnvType": "product",
      "OwnerIdList": {
        "OwnerIds": [
         88653
       ]
      "DbaId": 104442,
      "OwnerNameList": {
       "OwnerNames": [
          "dmstest"
      },
      "Port": 3306,
      "DatabaseId": 116366,
      "State": "NORMAL",
      "SearchName": "xxx test@pc-xxx.rwlb.rds.aliyuncs.com:3306 [xxx test] ",
      "DbaName": "xxx[子账号]",
      "SchemaName": "xxx test"
    }
  ]
}
```

6.6.4. ListLogicDatabases

调用ListLogicDatabases接口获取逻辑库详情列表。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListLogicDat abas es	系统规定参数。取 值:ListLogicDatabases。
PageNumber	Integer	是	1	页码。
Tid	Long	是	-1	租户ID。

名称	类型	是否必选	示例值	描述
PageSize	Integer	否	50	每页记录数。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
LogicDat abaseL ist	Array of LogicDatabase		逻辑库详情列表。
LogicDatabase			
Databaseld	String	1	逻辑库ID。
DbType	String	mysql	逻辑库数据库类型。
EnvType	String	test	逻辑库所属环境。
Logic	Boolean	true	是否逻辑库,固定为true。
OwnerldList	List	1	逻辑库Owner ID列表。
OwnerNameList	List	owner_name	逻辑库Owner名称列表。
SchemaName	String	logic_db	逻辑库名。
SearchName	String	loigc_db	逻辑库搜索名。
RequestId	String	BAD154C0-DA38- 48FA-97AB- E72912EEAA24	请求ID。
Success	Boolean	true	请求是否成功。

名称	类型	示例值	描述
TotalCount	Long	13	逻辑库总数。

示例

请求示例

http(s)://[Endpoint]/?Action=ListLogicDatabases
&PageNumber=1
&Tid=-1

&<公共请求参数>

正常返回示例

XML 格式

数据管理 开发指南·API参考

```
<TotalCount>13</TotalCount>
<RequestId>BAD154C0-DA38-48FA-97AB-E72912EEAA24/RequestId>
<LogicDatabaseList>
    <LogicDatabase>
        <Logic>true</Logic>
        <OwnerNameList>
           <OwnerNames>xxx</OwnerNames>
        </OwnerNameList>
        <DatabaseId>257</patabaseId>
        <SearchName>test db (我是逻辑库) </SearchName>
        <DbType>mysql</DbType>
        <OwnerIdList>
            <OwnerIds>88653</OwnerIds>
        </OwnerIdList>
        <EnvType>dev</EnvType>
        <SchemaName>test_db</SchemaName>
    </LogicDatabase>
    <LogicDatabase>
        <Logic>true</Logic>
        <OwnerNameList>
           <OwnerNames>xxx</OwnerNames>
           <OwnerNames>xxx</OwnerNames>
           <OwnerNames>xxx</OwnerNames>
        </OwnerNameList>
        <DatabaseId>258/DatabaseId>
        <SearchName>xxx_test (我是逻辑库) </SearchName>
        <DbType>mysql</DbType>
        <OwnerIdList>
            <OwnerIds>88656</OwnerIds>
            <OwnerIds>88653</OwnerIds>
           <OwnerIds>101252</OwnerIds>
        </OwnerIdList>
        <EnvType>product</EnvType>
        <SchemaName>idb_test</SchemaName>
    </LogicDatabase>
</LogicDatabaseList>
<Success>true</Success>
```

JSON 格式

```
"TotalCount": 13,
"RequestId": "BAD154C0-DA38-48FA-97AB-E72912EEAA24",
"LogicDatabaseList": {
  "LogicDatabase": [
      "Logic": true,
      "OwnerNameList": {
       "OwnerNames": [
         "xxx"
       ]
      },
      "DatabaseId": 257,
      "SearchName": "test_db (我是逻辑库) ",
      "DbType": "mysql",
      "OwnerIdList": {
       "OwnerIds": [
         88653
       ]
      "EnvType": "dev",
      "SchemaName": "test_db"
    },
      "Logic": true,
      "OwnerNameList": {
       "OwnerNames": [
         "xxx",
         "xxx",
         "xxx"
      },
      "DatabaseId": 258,
      "SearchName": "xxx_test (我是逻辑库)",
      "DbType": "mysql",
      "OwnerIdList": {
        "OwnerIds": [
         88656,
         88653,
         101252
      },
      "EnvType": "product",
      "SchemaName": "idb_test"
  ]
},
"Success": true
```

6.6.5. ListTables

调用ListTables接口获取数据库中表的详情列表。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListTables	系统规定参数。取值:ListTables。
Databaseld	String	是	1	数据库ID。
PageNumber	Integer	是	1	页码。
Tid	Long	是	-1	租户ID。
PageSize	Integer	否	10	每页记录数。
SearchName	String	否	test	表关键字,前缀匹配。
ReturnGuid	Boolean	否	true	是否返回表的GUID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	B16FB618-5E96- 4FFD-BB0D- 490C890A4030	请求ID。
Success	Boolean	true	请求是否成功。
TableList	Array of Table		表详情列表。
Table			
Databaseld	String	1	数据库ID。

名称	类型	示例值	描述
Description	String	test	表描述信息。
Encoding	String	utf8	表编码。
Engine	String	innodb	表引擎。
NumRows	Long	1	表行数(统计值,非精确行数)。
OwnerldList	List	1	表Owner ID列表。
OwnerNameList	List	owner_name	表Owner昵称列表。
StoreCapacity	Long	1024	表占用存储空间容量(统计值,非精确容量)。
TableGuid	String	IDB_308302.yuyang_ test.test_ch	表GUID。
TableId	String	1	表ID。
TableName	String	test	表名。
TableSchemaN ame	String	test_db	表所对应库名。
TableType	String	NORMAL	表类型,默认为NORMAL。
TotalCount	Long	12	满足条件的表总个数。

示例

请求示例

http(s)://[Endpoint]/?Action=ListTables &DatabaseId=1 &PageNumber=1 &Tid=-1 &<公共请求参数>

正常返回示例

XML 格式

```
<TableList>
   <Table>
       <TableSchemaName>yuyang test</TableSchemaName>
       <OwnerNameList>
            <OwnerNames>xxx</OwnerNames>
           <OwnerNames>xxx</OwnerNames>
       </OwnerNameList>
       <DatabaseId>116367</DatabaseId>
       <Encoding>utf8mb4</Encoding>
       <NumRows>3</NumRows>
       <StoreCapacity>1</StoreCapacity>
       <TableName>base user</TableName>
       <TableGuid>IDB 18396363.yuyang test.base user</TableGuid>
       <TableType>NORMAL</TableType>
       <OwnerIdList>
           <OwnerIds>88653</ownerIds>
            <OwnerIds>104442
       </OwnerIdList>
       <Engine>InnoDB</Engine>
       <TableId>18396363</TableId>
    </Table>
    <Table>
       <TableSchemaName>yuyang test</TableSchemaName>
       <OwnerNameList>
           <OwnerNames>xxx</OwnerNames>
           <OwnerNames>xxx</OwnerNames>
       </OwnerNameList>
       <DatabaseId>116367/DatabaseId>
       <Encoding>utf8mb4</Encoding>
       <NumRows>2</NumRows>
       <StoreCapacity>1</StoreCapacity>
       <TableName>biz_orders</TableName>
        <TableType>NORMAL</TableType>
       <OwnerIdList>
           <OwnerIds>88653</OwnerIds>
           <OwnerIds>104442
       </OwnerIdList>
       <Engine>InnoDB</Engine>
       <TableId>18390459</TableId>
    </Table>
</TableList>
<TotalCount>12</TotalCount>
<RequestId>B16FB618-5E96-4FFD-BB0D-490C890A4030/RequestId>
<Success>true</Success>
```

JSON 格式

```
"TableList": {
    "Table": [
      {
         "TableSchemaName": "yuyang_test",
         "OwnerNameList": {
```

```
"OwnerNames": [
         "xxx",
          "xxx"
       ]
      "DatabaseId": 116367,
      "Encoding": "utf8mb4",
      "NumRows": 3,
      "StoreCapacity": 1,
      "TableName": "base user",
      "TableGuid": "IDB_18396363.yuyang_test.base_user",
      "TableType": "NORMAL",
      "OwnerIdList": {
       "OwnerIds": [
         88653,
         104442
      1
      "Engine": "InnoDB",
      "TableId": 18396363
    },
      "TableSchemaName": "yuyang test",
      "OwnerNameList": {
       "OwnerNames": [
         "xxx",
         "xxx"
       ]
      },
      "DatabaseId": 116367,
      "Encoding": "utf8mb4",
      "NumRows": 2,
      "StoreCapacity": 1,
      "TableName": "biz orders",
      "TableType": "NORMAL",
      "OwnerIdList": {
       "OwnerIds": [
         88653,
         104442
       ]
      "Engine": "InnoDB",
      "TableId": 18390459
 ]
"TotalCount": 12,
"RequestId": "B16FB618-5E96-4FFD-BB0D-490C890A4030",
"Success": true
```

6.6.6. ListLogicTables

调用ListLogicTables接口获取指定逻辑库下面的逻辑表详情列表。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListLogicTables	系统规定参数。取 值:ListLogicTables。
Databaseld	String	是	1	逻辑库ID。
PageNumber	Integer	是	1	页码。
Tid	Long	是	-1	租户ID。
PageSize	Integer	否	10	每页记录数。
SearchName	String	否	test	逻辑表搜索关键字,前缀匹配。
ReturnGuid	Boolean	否	true	是否返回表的GUID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
LogicTableList	Array of LogicTable		逻辑表详情列表。
LogicTable			
Databaseld	String	1	逻辑库ID。
Logic	Boolean	true	是否为逻辑表,固定为true。
OwnerldList	List	1	逻辑表Owner ID列表。

名称	类型	示例值	描述
OwnerNameList	List	owner_name	逻辑表Owner名称列表。
SchemaName	String	yuyang_test	逻辑表对应逻辑库名。
TableCount	String	4	分表个数。
TableExpr	String	test[1-4]	逻辑表表达式。
TableGuid	String	IDB_L_308302.yuyan g_test.test_ch	逻辑表GUID。
TableId	String	1	逻辑表ID。
TableName	String	test	逻辑表表名。
RequestId	String	F1E6484F-9DF1- 4406-9BDE- 0861C4629B69	请求ID。
Success	Boolean	true	请求是否成功。
TotalCount	Long	1	满足条件记录数。

示例

请求示例

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

&DatabaseId=1

&PageNumber=1

&Tid=-1

&<公共请求参数>

正常返回示例

XML 格式

数据管理 开发指南·API参考

```
<TotalCount>1</TotalCount>
<RequestId>F1E6484F-9DF1-4406-9BDE-0861C4629B69/RequestId>
<LogicTableList>
   <LogicTable>
       <Logic>true</Logic>
       <OwnerNameList>
           <OwnerNames>xxx</OwnerNames>
           <OwnerNames>xxx</OwnerNames>
       </OwnerNameList>
       <TableExpr>t[1-4]</TableExpr>
       <DatabaseId>267</DatabaseId>
       <SchemaName>yuyang_test</SchemaName>
       <TableName>t</TableName>
       <TableGuid>IDB_L_3057.yuyang_test.t</TableGuid>
       <TableCount>4</TableCount>
       <OwnerIdList>
           <OwnerIds>88653</OwnerIds>
           <OwnerIds>104442
       </OwnerIdList>
       <TableId>3057</TableId>
    </LogicTable>
</LogicTableList>
<Success>true</Success>
```

JSON 格式

```
"TotalCount": 1,
"RequestId": "F1E6484F-9DF1-4406-9BDE-0861C4629B69",
"LogicTableList": {
  "LogicTable": [
      "Logic": true,
      "OwnerNameList": {
       "OwnerNames": [
         "xxx",
         "xxx"
       ]
      },
      "TableExpr": "t[1-4]",
      "DatabaseId": 267,
      "SchemaName": "yuyang_test",
      "TableName": "t",
      "TableGuid": "IDB_L_3057.yuyang_test.t",
      "TableCount": 4,
      "OwnerIdList": {
       "OwnerIds": [
         88653,
        104442
       ]
      },
      "TableId": 3057
 ]
},
"Success": true
```

6.6.7. ListColumns

调用List Columns接口获取表的列信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListColumns	系统规定参数。取值:List Columns。
Logic	Boolean	是	false	是否逻辑表。
TableId	String	是	1	表ID。
Tid	Long	是	0	租户ID。

名称	类型	是否必选	示例值	描述
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ColumnList	Array of Column		列信息详情。
Column			
AutoIncrement	Boolean	false	当前字段是否自增。取值: ● true: 自增。 ● false: 不自增。
Columnid	String	1	字段ID。
ColumnName	String	name	字段名。
ColumnType	String	varchar	字段类型。
DataLength	Long	32	字段长度。
DataPrecision	Integer	0	字段精度。
DataScale	Integer	0	字段精度。
DefaultValue	String	def_value	字段默认值。
Description	String	test	字段描述。
FunctionType	String	default	字段遮掩算法类型,取值如下: null: 无遮掩 DEFAULT: 全遮掩 FIX_POS: 固定位置遮掩 FIX_CHAR: 固定字符替换
Nullable	Boolean	true	字段是否可空。

名称	类型	示例值	描述
SecurityLevel	String	INNER	安全级别,取值如下: INNER: 内部(非敏感) SENSITIVE: 敏感 CONFIDENTIAL: 机密
Sensitive	Boolean	false	是否敏感字段。
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	0AD9AC55-5873- 474A-9F33- 4285806A3619	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=ListColumns
&Logic=false
&TableId=1
&Tid=0

&<公共请求参数>

正常返回示例

XML 格式

```
<ColumnList>
   <Column>
       <ColumnId>314036829</ColumnId>
       <DataScale>0</DataScale>
       <Sensitive>false</Sensitive>
       <Description>主键ID</Description>
       <DataPrecision>20</DataPrecision>
       <ColumnType>bigint(20) unsigned</ColumnType>
       <ColumnName>id</ColumnName>
       <SecurityLevel>INNER</SecurityLevel>
       <AutoIncrement>true</AutoIncrement>
       <Nullable>false</Nullable>
   </Column>
   <Column>
       <ColumnId>314036830</ColumnId>
       <Sensitive>false</Sensitive>
       <Description>创建时间
       <DefaultValue>CURRENT TIMESTAMP/DefaultValue>
       <ColumnType>datetime</ColumnType>
       <ColumnName>gmt create</ColumnName>
       <SecurityLevel>INNER</SecurityLevel>
       <AutoIncrement>false</AutoIncrement>
       <Nullable>false</Nullable>
   </Column>
    <Column>
       <ColumnId>314036831</ColumnId>
       <Sensitive>false</Sensitive>
       <Description>修改时间
       <DefaultValue>CURRENT TIMESTAMP</DefaultValue>
       <ColumnType>datetime</ColumnType>
       <ColumnName>gmt modified</ColumnName>
       <SecurityLevel>INNER</SecurityLevel>
       <AutoIncrement>false</AutoIncrement>
       <Nullable>false</Nullable>
   </Column>
   <Column>
       <ColumnId>314036832</ColumnId>
       <Sensitive>false</Sensitive>
       <Description>base user/Description>
       <ColumnType>varchar(64)</ColumnType>
       <DataLength>64/DataLength>
       <ColumnName>user name</ColumnName>
       <SecurityLevel>INNER</SecurityLevel>
       <AutoIncrement>false/AutoIncrement>
       <Nullable>true</Nullable>
   </Column>
</ColumnList>
<RequestId>0AD9AC55-5873-474A-9F33-4285806A3619/RequestId>
<Success>true</Success>
```

JSON 格式

```
"ColumnList": {
  "Column": [
      "ColumnId": 314036829,
      "DataScale": 0,
      "Sensitive": false,
      "Description": "主键ID",
      "DataPrecision": 20,
      "ColumnType": "bigint(20) unsigned",
      "ColumnName": "id",
      "SecurityLevel": "INNER",
      "AutoIncrement": true,
      "Nullable": false
    },
      "ColumnId": 314036830,
      "Sensitive": false,
      "Description": "创建时间",
      "DefaultValue": "CURRENT TIMESTAMP",
      "ColumnType": "datetime",
      "ColumnName": "gmt_create",
      "SecurityLevel": "INNER",
      "AutoIncrement": false,
      "Nullable": false
    },
      "ColumnId": 314036831,
      "Sensitive": false,
      "Description": "修改时间",
      "DefaultValue": "CURRENT TIMESTAMP",
      "ColumnType": "datetime",
      "ColumnName": "gmt modified",
      "SecurityLevel": "INNER",
      "AutoIncrement": false,
      "Nullable": false
      "ColumnId": 314036832,
      "Sensitive": false,
      "Description": "base user",
      "ColumnType": "varchar(64)",
      "DataLength": 64,
      "ColumnName": "user name",
      "SecurityLevel": "INNER",
      "AutoIncrement": false,
      "Nullable": true
   }
 ]
},
"RequestId": "0AD9AC55-5873-474A-9F33-4285806A3619",
"Success": true
```

数据管理 开发指南·API参考

6.6.8. ListIndexes

调用ListIndexes接口返回表索引信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListIndexes	系统规定参数。取值:ListIndexes。
Logic	Boolean	是	false	是否为逻辑表。
TableId	String	是	1	表ID。
Tid	Long	是	0	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
IndexList	Array of Index		索引详情列表。
Index			
IndexComment	String	test	索引描述。
IndexId	String	1	索引ID。
IndexName	String	idx_test	索引名称。

名称	类型	示例值	描述
IndexType	String	Primary	索引类型,取值如下: Primary Unique Normal FullText Spatial
TableId	String	1	表ID。
RequestId	String	1F4DE2F1-5B47- 462A-A973- E02EB7AF386B	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ListIndexes
&Logic=false
&TableId=1
&Tid=0
&<公共请求参数>
```

正常返回示例

XML 格式

JSON 格式

数据管理 开发指南·API参考

6.6.9. GetDatabase

调用GetDatabase接口获取具体库的详情信息。

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 。
Host	String	是	xxx.xxx.xxx	实例连接地址。
Port	Integer	是	3306	实例连接端口。
SchemaName	String	是	test_db	数据库名。
Tid	Long	是	-1	租户ID。
Sid	String	否	test_sid	实例SID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
Database	Struct		库详情信息。

名称	类型	示例值	描述
CatalogName	String	def	数据库目录名。 ② 说明 若数据库类型为PG系列,此处表示PG数据库名。
Databaseld	String	1	数据库ID。
DbType	String	mysql	数据库类型。
Dbald	String	1	DBA ID。
DbaName	String	dba_name	DBA名称。
Encoding	String	utf8mb4	数据库编码。
EnvType	String	test	库环境类型。
Host	String	XXX.XXX.XXX	实例连接地址。
Instanceld	String	1	实例ID。
OwnerldList	List	1	数据库Owner ID列表。
OwnerNameList	List	test	数据库Owner名称列表。
Port	Integer	3306	实例连接端口。
SchemaName	String	test_db	Schema名称。
SearchName	String	test_db@xxx.xxx.xxx .xxx:3306	逻辑库搜索名。
Sid	String	test_sid	实例SID。
State	String	NORMAL	库状态,取值参考List Dat abases。

数据管理 开发指南·API参考

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	3CDB8601-AD74- 4A47-8114- 08E08CD6319D	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=GetDatabase
&Host=xxx.xxx.xxx
&Port=3306
&SchemaName=test_db
&Tid=-1

&<公共请求参数>

正常返回示例

XML 格式

```
<Database>
   <Encoding>utf8mb4</Encoding>
   <Host>pc-xxx.rwlb.rds.aliyuncs.com
   <InstanceId>24710</InstanceId>
   <DbType>polardb</pbType>
   <OwnerIdList>
       <OwnerIds>88653</OwnerIds>
       <OwnerIds>104442</OwnerIds>
   </OwnerIdList>
   <EnvType>product</EnvType>
   <DbaId>104442
   <OwnerNameList>
       <OwnerNames>xxx</OwnerNames>
       <OwnerNames>xxx[子账号]</OwnerNames>
   </OwnerNameList>
   <Port>3306</Port>
   <DatabaseId>116367</DatabaseId>
   <State>NORMAL</State>
   <SearchName>xxx test@pc-xxx.rwlb.rds.aliyuncs.com:3306 [xxx test] 
   <DbaName>xxx[子账号]</DbaName>
   <SchemaName>xxx_test
</Database>
<RequestId>3CDB8601-AD74-4A47-8114-08E08CD6319D/RequestId>
<Success>true</Success>
```

JSON 格式

```
"Database": {
  "Encoding": "utf8mb4",
  "Host": "pc-xxx.rwlb.rds.aliyuncs.com",
  "InstanceId": 24710,
  "DbType": "polardb",
  "OwnerIdList": {
    "OwnerIds": [
     88653,
     104442
  "EnvType": "product",
  "DbaId": 104442,
  "OwnerNameList": {
   "OwnerNames": [
     "xxx",
     "xxx[子账号]"
  "Port": 3306,
  "DatabaseId": 116367,
  "State": "NORMAL",
  "SearchName": "xxx_test@pc-xxx.rwlb.rds.aliyuncs.com:3306 [xxx_test] ",
  "DbaName": "xxx[子账号]",
  "SchemaName": "xxx_test"
"RequestId": "3CDB8601-AD74-4A47-8114-08E08CD6319D",
"Success": true
```

6.6.10. GetLogicDatabase

调用GetLogicDatabase接口获取逻辑库详情信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetLogicDatabas e	系统规定参数。取 值:GetLogicDatabase。
Dbld	String	是	1	逻辑库DB ID。
Tid	Long	是	0	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
LogicDatabase	Struct		逻辑库详情信息。
Databaseld	String	1	逻辑库ID。
DbType	String	mysql	数据库类型。
EnvType	String	dev	库环境类型。
Logic	Boolean	true	是否逻辑库,固定为true。
OwnerldList	List	1	库Owner ID列表。
OwnerNameList	List	owner_name	库Owner名称列表。
SchemaName	String	test_logic_db	逻辑库库名。
SearchName	String	test_logic_db[xxx]	逻辑库搜索名。
RequestId	String	90260530-565C- 42B9-A6E8- 893481FE6AB6	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=GetLogicDatabase
&DbId=1
&Tid=0

&<公共请求参数>

正常返回示例

XML 格式

```
<LogicDatabase>
   <Logic>true</Logic>
   <OwnerNameList>
       <OwnerNames>xxx</OwnerNames>
       <OwnerNames>xxx</OwnerNames>
       <OwnerNames>xxx</OwnerNames>
   </OwnerNameList>
   <DatabaseId>265/DatabaseId>
   <SearchName>test_logic_db(xxx)
   <DbType>mysql</DbType>
   <OwnerIdList>
       <OwnerIds>88653</OwnerIds>
       <OwnerIds>97787</OwnerIds>
       <OwnerIds>88654
   </OwnerIdList>
   <EnvType>dev</EnvType>
   <SchemaName>test_logic_db</SchemaName>
</LogicDatabase>
<RequestId>90260530-565C-42B9-A6E8-893481FE6AB6</RequestId>
<Success>true</Success>
```

JSON 格式

```
"LogicDatabase": {
  "Logic": true,
  "OwnerNameList": {
   "OwnerNames": [
     "xxx",
      "xxx",
      "xxx"
   ]
  "DatabaseId": 265,
  "SearchName": "test logic db(xxx)",
  "DbType": "mysql",
  "OwnerIdList": {
   "OwnerIds": [
     88653,
     97787,
     88654
   ]
  "EnvType": "dev",
  "SchemaName": "test_logic_db"
"RequestId": "90260530-565C-42B9-A6E8-893481FE6AB6",
"Success": true
```

6.6.11. GetMetaTableDetailInfo

调用GetMetaTableDetailInfo接口获取表的字段、索引等信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetMetaTableDet ailInfo	系统规定参数。取 值:GetMetaTableDetailInfo。
TableGuid	String	是	IDB_L_9032.db- test.yuyang_test	表在DMS中的唯一识别码GUID(Globally Unique Identifier)。 ② 说明 ● 逻辑库: 可调 用ListLogicTables接口获取,需要将ReturnGuid参数设置为true。 ● 物理库: 可调 用ListTables接口获取,需要将ReturnGuid参数设置为true。
Tid	Long	否	123	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
DetailInfo	Struct		表详情。
ColumnList	Array of Column		字段列表。

名称	类型	示例值	描述
AutoIncrement	Boolean	true	当前字段是否自增。取值: ● true: 自增。 ● false: 不自增。
Columnid	String	191234849	字段ID。
ColumnName	String	id	字段名。
ColumnType	String	bigint (20) unsigned	字段的数据类型。例 如:Bigint、Int、Varchar。
DataLength	Integer	0	字段长度。
DataPrecision	Integer	0	字段精度。
DataScale	Integer	0	该字段小数后的位数。
Description	String	test	字段描述。
Nullable	Boolean	false	是否可空。取值: • true: 允许为空。 • false: 不允许为空。
Position	String	1	字段在表中的顺序。
IndexList	Array of Index		索引列表。
IndexColumns	List	id,name	索引字段。
IndexId	String	123	索引ID。
IndexName	String	PRIMARY	索引名。
IndexType	String	Primary	索引类型。例 如:Primary、Unique、Normal。

名称	类型	示例值	描述
Unique	Boolean	false	是否为唯一索引。取值: • true: 是唯一索引。 • false: 不是唯一索引。
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	E881CB2F-DE42- 42E5-90EB- 8B3173DCB9B9	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=GetMetaTableDetailInfo &TableGuid=IDB_L_9032.db-test.yuyang_test &<公共请求参数>

正常返回示例

XML 格式

数据管理 开发指南·API参考

```
<RequestId>E881CB2F-DE42-42E5-90EB-8B3173DCB9B9/RequestId>
<DetailInfo>
   <ColumnList>
       <ColumnId>191234849</ColumnId>
       <ColumnName>id</ColumnName>
       <Description>test/Description>
       <ColumnType>bigint(20) unsigned</ColumnType>
       <Position>1</Position>
       <AutoIncrement>true</AutoIncrement>
       <DataScale>0</DataScale>
       <DataPrecision>20</DataPrecision>
       <Nullable>false</Nullable>
   </ColumnList>
   <ColumnList>
       <ColumnId>191234850</ColumnId>
       <ColumnName>gmt_create</ColumnName>
       <Description>test</Description>
       <ColumnType>datetime</ColumnType>
       <Position>2</Position>
       <AutoIncrement>false</AutoIncrement>
       <Nullable>false</Nullable>
   </ColumnList>
   <ColumnList>
       <ColumnId>191234851</ColumnId>
       <ColumnName>gmt modified</ColumnName>
       <Description>test/Description>
       <ColumnType>datetime</ColumnType>
       <Position>3</Position>
       <AutoIncrement>false</AutoIncrement>
       <Nullable>false</Nullable>
   </ColumnList>
   <IndexList>
       <IndexName>PRIMARY</IndexName>
       <IndexId>33008763</IndexId>
       <IndexColumns>id</IndexColumns>
       <Unique>false</Unique>
       <IndexType>Primary</IndexType>
   </IndexList>
</DetailInfo>
<Success>true</Success>
```

JSON 格式

```
"RequestId": "E881CB2F-DE42-42E5-90EB-8B3173DCB9B9",
"DetailInfo": {
  "ColumnList": [
      "ColumnId": 191234849,
      "ColumnName": "id",
      "Description": "test",
      "ColumnType": "bigint(20) unsigned",
      "Position": 1,
      "AutoIncrement": true,
      "DataScale": 0,
      "DataPrecision": 20,
      "Nullable": false
    },
      "ColumnId": 191234850,
      "ColumnName": "gmt create",
      "Description": "test",
      "ColumnType": "datetime",
      "Position": 2,
      "AutoIncrement": false,
      "Nullable": false
    },
      "ColumnId": 191234851,
      "ColumnName": "gmt modified",
      "Description": "test",
      "ColumnType": "datetime",
      "Position": 3,
      "AutoIncrement": false,
      "Nullable": false
   }
  ],
  "IndexList": [
      "IndexName": "PRIMARY",
      "IndexId": 33008763,
      "IndexColumns": [
       "id"
      "Unique": false,
      "IndexType": "Primary"
    }
},
"Success": true
```

6.6.12. GetTableDBTopology

调用GetTableDBTopology接口获取数据表的拓扑图。

目前仅支持<mark>逻辑库</mark>实例调用该API。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetTableDBTopol ogy	系统规定参数。取 值:GetTableDBTopology。
T ableGuid	String	是	IDB_L_9032.db- test.yuyang_test	表在DMS中的唯一识别码GUID (Globally Unique Identifier)。 ② 说明 ● 逻辑库: 可调 用ListLogicTables接口获取,需要将ReturnGuid参数设置为true。 ● 物理库: 可条 用ListTables接口获取,需要将ReturnGuid参数设置为true。
Tid	Long	否	123	租户ID。
RegionId	String	否	cn-hangzhou	选择DMS的服务地域,用于调用该接口。

返回数据

名称	类型	示例值	描述
DBTopology	Struct		数据库拓扑信息。
DataSourceList	Array of DataSource		数据源列表。
DatabaseList	Array of Database		物理分库列表。
Dbld	String	489347	数据库ID。
DbName	String	db-test	物理库库名。

名称	类型	示例值	描述
DbType	String	mysql	数据库类型。
EnvType	String	pre	环境类型,取值和说明如下: product: 生产环境 dev: 开发环境 pre: 预发环境 test: 测试环境 sit: SIT环境 uat: UAT环境 pet: 压测环境 stag: STAG环境
TableList	Array of Table		物理分表列表。
TableId	String	151977812	表ID。
TableName	String	yuyang_test_0000	物理表表名。
TableType	String	NORMAL	表类型,保留字段。
DbType	String	mysql	数据库类型。
Host	String	xxx.mysql.polardb.r ds.aliyuncs.com	数据源的连接地址。
Port	Integer	3306	数据源连接端口。
Sid	String	def	数据源SID。
TableGuid	String	IDB_L_9032.db- test.yuyang_test	表在DMS中的唯一识别码GUID(Globally Unique Identifier)。

数据管理 开发指南·API参考

名称	类型	示例值	描述
			表名。
TableName	String	yuyang_test	⑦ 说明● 如果查询逻辑表,将返回逻辑表表名。● 如果查询物理表,将返回物理表表名。
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	853F7FD4-D922- 4EFB-931C- D253EF159E06	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=GetTableDBTopology &TableGuid=IDB_L_9032.db-test.yuyang_test &<**公共请求参数**>

正常返回示例

XML 格式

```
<RequestId>55B7CF50-4610-4707-B2B2-5EF2BE000367/RequestId>
<DBTopology>
   <TableName>yuyang test</TableName>
   <TableGuid>IDB_L_9032.db-test.yuyang_test</TableGuid>
   <DataSourceList>
        <DatabaseList>
           <DbId>489347</pbId>
            <DbName>db-test</DbName>
            <TableList>
                <TableId>151977811</TableId>
                <TableName>yuyang test 0000</TableName>
            </TableList>
            <TableList>
                <TableId>151977812</TableId>
                <TableName>yuyang_test_0001</TableName>
            </TableList>
            <TableList>
                <TableId>151977813</TableId>
                <TableName>yuyang test 0002</TableName>
            </TableList>
            <TableList>
                <TableId>151977814</TableId>
                <TableName>yuyang test 0003</TableName>
            </TableList>
            <TableList>
                <TableId>151977815</TableId>
                <TableName>yuyang_test_0004</TableName>
            </TableList>
            <EnvType>dev</EnvType>
            <DbType>polardb</DbType>
        </DatabaseList>
        <Port>3306</Port>
        <Host>xxx.mysql.polardb.rds.aliyuncs.com/Host>
        <DbType>polardb</pbType>
    </DataSourceList>
</DBTopology>
<Success>true</Success>
```

JSON 格式

```
"RequestId": "55B7CF50-4610-4707-B2B2-5EF2BE000367",
"DBTopology": {
  "TableName": "yuyang_test",
  "TableGuid": "IDB L 9032.db-test.yuyang_test",
  "DataSourceList": [
      "DatabaseList": [
          "DbId": 489347,
          "DbName": "db-test",
          "TableList": [
              "TableId": 151977811,
              "TableName": "yuyang_test_0000"
            },
              "TableId": 151977812,
              "TableName": "yuyang test 0001"
            },
              "TableId": 151977813,
              "TableName": "yuyang test 0002"
            },
              "TableId": 151977814,
              "TableName": "yuyang test 0003"
            },
              "TableId": 151977815,
              "TableName": "yuyang test 0004"
          "EnvType": "dev",
          "DbType": "polardb"
      ],
      "Port": 3306,
      "Host": "xxx.mysql.polardb.rds.aliyuncs.com",
      "DbType": "polardb"
  ]
},
"Success": true
```

6.6.13. GetMetaTableColumn

调用GetMetaTableColumn接口获取表的字段信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetMetaTableCol umn	系统规定参数。取 值:GetMetaTableColumn。
TableGuid	String	是	IDB_L_11345.yuya ng_test.base_use r	表在DMS中的唯一识别码GUID (Globally Unique Identifier)。 ② 说明 • 逻辑库: 可调 用ListLogicTables接口获取,需要将ReturnGuid参数设置为true。 • 物理库: 可调 用ListTables接口获取,需要将ReturnGuid参数设置为true。
Tid	Long	否	123	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ColumnList	Array of Column		字段列表。
AutoIncrement	Boolean	true	字段是否自增。取值: ● true: 自增。 ● false: 不自增。
Columnid	String	191234849	字段ID。
ColumnName	String	id	字段名。
ColumnType	String	bigint (20) unsigned	字段的数据类型。例如:Bigint 、Int、Varchar。
DataLength	Integer	0	字段长度。

名称	类型	示例值	描述
DataPrecision	Integer	0	字段精度。
DataScale	Integer	0	字段小数后的位数。
Description	String	test	字段描述。
Nullable	Boolean	false	字段是否允许为空。取值: • true:允许为空。 • false:不允许为空。
Position	Integer	1	字段在表中顺序。
PrimaryKey	String	true	字段是否为主键,取值: • true: 主键。 • false: 非主键。
SecurityLevel	String	INNER	字段的安全级别,取值如下: INNER: 内部(非敏感) SENSITIVE: 敏感 CONFIDENTIAL: 机密
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	2B581A5E-D4FE- 4F77-BADE- DD944D7BEBE2	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=GetMetaTableColumn &TableGuid=IDB_L_11345.yuyang_test.base_user &<公共请求参数>

正常返回示例

XML 格式

```
<GetMetaTableColumnResponse>
     <RequestId>2B581A5E-D4FE-4F77-BADE-DD944D7BEBE2/RequestId>
     <ErrorCode>UnknownError
           <ColumnId>191234849</ColumnId>
           <ColumnName>id</ColumnName>
           <Description>test/Description>
           <ColumnType>bigint(20) unsigned</ColumnType>
           <Position>1</Position>
           <AutoIncrement>true</AutoIncrement>
           <SecurityLevel>INNER</SecurityLevel>
           <DataScale>0</DataScale>
           <DataLength>0</DataLength>
           <DataPrecision>0</DataPrecision>
           <PrimaryKey>true</PrimaryKey>
           <Nullable>false</Nullable>
     </ColumnList>
     <ErrorMessage>UnknownError</ErrorMessage>
     <Success>true</Success>
</GetMetaTableColumnResponse>
```

JSON 格式

```
"RequestId": "2B581A5E-D4FE-4F77-BADE-DD944D7BEBE2",
"ErrorCode": "UnknownError",
"ColumnList": [{
"ColumnId": "191234849",
"ColumnName": "id",
"Description": "test",
"ColumnType": "bigint(20) unsigned",
"Position": "1",
 "AutoIncrement": "true",
 "SecurityLevel": "INNER",
"DataScale": "0",
"DataLength": "0",
"DataPrecision": "0",
"PrimaryKey": "true",
"Nullable": "false"
}],
"ErrorMessage": "UnknownError",
"Success": "true"
```

6.6.14. SyncDatabaseMeta

调用SyncDatabaseMeta接口同步数据库字典元数据。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	SyncDat abaseMet a	系统规定参数。取 值:SyncDatabaseMeta。
Dbld	String	是	123	数据库ID。
Tid	Long	是	-1	租户ID。
Logic	Boolean	否	false	是否为逻辑库。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	A99CD576-1E18- 4E86-931E- C3CCE56DC030	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=SyncDatabaseMeta
&DbId=123

&Tid=-1

&<公共请求参数>

正常返回示例

XML 格式

<RequestId>A99CD576-1E18-4E86-931E-C3CCE56DC030</RequestId><Success>true</Success>

JSON 格式

```
{
   "RequestId": "A99CD576-1E18-4E86-931E-C3CCE56DC030",
   "Success": true
}
```

6.6.15. SyncInstanceMeta

调用SyncInstanceMeta接口同步实例下所有数据库字典元数据。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	SyncInstanceMeta	系统规定参数,取 值:SyncInstanceMeta。
InstanceId	String	是	1234	实例ID。
Tid	Long	是	-1	租户ID。
IgnoreTable	Boolean	否	true	是否忽略表字典元数据同步。取值: true false
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误消息。
RequestId	String	A99CD576-1E18- 4E86-931E- C3CCE56DC030	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

 $\label{eq:http} $$ $$ \true (s): $$ $$ $$ //[Endpoint]/?Action=SyncInstanceMeta $$ $$ \creen support $$ $$ $$ $$ $$ $$ $$$

正常返回示例

XML 格式

<RequestId>A99CD576-1E18-4E86-931E-C3CCE56DC030</RequestId>
<Success>true</Success>

JSON 格式

```
{
   "RequestId": "A99CD576-1E18-4E86-931E-C3CCE56DC030",
   "Success": true
}
```

6.7. 工单操作

6.7.1. 通用

6.7.1.1. ApproveOrder

调用ApproveOrder接口审批工单。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ApproveOrder	系统规定参数。取值: ApproveOrder 。
ApprovalType	String	是	agree	审批动作,取值和说明如下: ● AGREE: 同意 ● CANCEL: 取消 ● REJECT: 拒绝
Tid	Long	是	-1	租户ID。 ② 说明 DMS控制台右上角头像处悬停鼠标即可展示租户ID信息。
Workflowinstanc eld	Long	是	1234	审批流ID,该值可 从Get OrderBaseInf o 接口获取,针对需 要审批的工单,此值是大于0的整数。

名称	类型	是否必选	示例值	描述
RegionId	String	否	cn-hangzhou	地域ID。
Comment	String	否	test	备注信息。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ApproveOrder
&ApprovalType=agree
&Tid=-1
&WorkflowInstanceId=1234
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>427688B8-ADFB-4C4E-9D45-EF5C1FD6E23D</RequestId>
<Success>true</Success>
```

JSON 格式

```
{
    "RequestId": "427688B8-ADFB-4C4E-9D45-EF5C1FD6E23D",
    "Success": true
}
```

6.7.1.2. CloseOrder

调用CloseOrder接口关闭工单。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CloseOrder	系统规定参数。取值:CloseOrder。
CloseReason	String	是	close reason	关闭原因。
Orderld	Long	是	1343	工单号。
Tid	Long	是	-1	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。
Success	Boolean	true	操作是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=CloseOrder
&CloseReason=close reason

&OrderId=1343

&Tid=-1

&<公共请求参数>

正常返回示例

XML 格式

<RequestId>427688B8-ADFB-4C4E-9D45-EF5C1FD6E23D</RequestId><Success>true</Success>

JSON 格式

```
{
    "RequestId": "427688B8-ADFB-4C4E-9D45-EF5C1FD6E23D",
    "Success": true
}
```

6.7.1.3. GetApprovalDetail

调用GetApprovalDetail接口获取工单审批详情。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetApprovalDetai l	系统规定参数。取 值:GetApprovalDetail。
Tid	Long	是	-1	租户ID。
Workflowinstanc eid	Long	是	12345	审批流ID。您可以通过Get OrderBaseInf o接口获取该参数,仅针对需要审批的工单,此值是大于0的整数。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ApprovalDetail	Struct		审批详情信息。
AuditId	Long	277	审批流ID。
Current Handler s	Array of CurrentHandler		当前审批处理人信息。
CurrentHandler			
Id	Long	1	用户ID。

名称	类型	示例值	描述
NickName	String	nickName	用户名称。
Description	String	test	审批描述信息。
Orderld	Long	1234	工单ID。
OrderType	String	DATA_EXPORT	工单类型。
ReasonList	List	["reason1"]	审批原因。
Title	String	数据导出工单-1234	审批标题。
WorkflowInsCo de	String	APPROVED	审批状态码。
WorkflowNode s	Array of WorkflowNode		审批流程节点信息。
WorkflowNode			
AuditUserldList	List	104442	审批人ID列表。
NodeName	String	DBA	审批节点名称。
OperateComme nt	String	理由: 同意	审批操作备注。
OperateTime	String	2019-10-15 13:47:54	审批操作时间。
OperatorId	Long	13434	审批操作人ID。
WorkflowInsCo de	String	APPROVED	审批状态码。
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。

名称	类型	示例值	描述
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetApprovalDetail
&Tid=-1
&WorkflowInstanceId=12345
&<公共请求参数>
```

正常返回示例

XML 格式

```
<ApprovalDetail>
   <Description>xxx</Description>
   <WorkflowInsCode>APPROVED</WorkflowInsCode>
   <CurrentHandlers>
   </CurrentHandlers>
   <OrderId>737047</OrderId>
   <OrderType>DATA EXPORT
   <ReasonList>
       <Reasons>xxx</Reasons>
   </ReasonList>
   <AuditId>277</AuditId>
   <Title>数据导出工单-737047</Title>
   <WorkflowNodes>
       <WorkflowNode>
           <OperatorId>104442
          <WorkflowInsCode>APPROVED</WorkflowInsCode>
          <OperateComment>理由: 同意
          <AuditUserIdList>
              <AuditUserIds>104442</AuditUserIds>
          </AuditUserIdList>
          <OperateTime>2019-10-15 13:47:54
           <NodeName>DBA</NodeName>
       </WorkflowNode>
   </WorkflowNodes>
</ApprovalDetail>
<RequestId>FE8EE2F1-4880-46BC-A704-5CF63EAF9A04/RequestId>
<Success>true</Success>
```

JSON 格式

```
"ApprovalDetail": {
 "Description": "xxx",
 "WorkflowInsCode": "APPROVED",
  "CurrentHandlers": {
   "CurrentHandler": []
 },
  "OrderId": 737047,
  "OrderType": "DATA_EXPORT",
  "ReasonList": {
   "Reasons": [
     "xxx"
  },
  "AuditId": 277,
  "Title": "数据导出工单-737047",
  "WorkflowNodes": {
   "WorkflowNode": [
       "OperatorId": 104442,
       "WorkflowInsCode": "APPROVED",
       "OperateComment": "理由: 同意",
       "AuditUserIdList": {
         "AuditUserIds": [
           104442
        ]
       "OperateTime": "2019-10-15 13:47:54",
        "NodeName": "DBA"
   ]
 }
"RequestId": "FE8EE2F1-4880-46BC-A704-5CF63EAF9A04",
"Success": true
```

6.7.1.4. GetOrderBaseInfo

调用Get OrderBaseInf o接口获取工单基本信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	Get OrderBaseInfo	系统规定参数。取 值:GetOrderBaseInfo。
Orderld	Long	是	12345	工单ID。

名称	类型	是否必选	示例值	描述
Tid	Long	是	-1	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
OrderBaseInfo	Struct		工单基本信息。
Comment	String	test	备注信息。
Committer	String	xxx	提交者。
CommitterId	Long	1	提交者ID,注意不是提交者的阿里云UID。
CreateTime	String	2019-10-10 00:00:00	工单创建时间。
LastModifyTim e	String	2019-10-10 00:00:00	工单最后修改时间。
Orderld	Long	12345	工单号。
PluginType	String	DC_COMMON	工单类型,取值和说明参考 CreateOrder 接口相对应入参说明。
RelatedUserList	List	1,2,3	工单相关人ID列表。
RelatedUserNic kList	List	a,b,c	工单相关人名称列表。

数据管理 开发指南·API参考

名称	类型	示例值	描述
StatusCode	String	success	工单状态码,取值和说明如下: new:新建 toaudit:审批中 approved:审批通过 reject:审批拒绝 processing:执行中 success:执行成功 closed:已关闭
StatusDesc	String	success	状态描述信息。
WorkflowInstan celd	Long	1	审批流ID。
WorkflowStatu sDesc	String	approved	审批描述信息。
RequestId	String	7133DF67-5B25- 460F-8285- C4CC93472C2F	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=GetOrderBaseInfo
&OrderId=12345

&Tid=-1

&<公共请求参数>

正常返回示例

XML 格式

```
<OrderBaseInfo>
   <Comment>test</Comment>
   <CreateTime>2019-10-10 00:00:00
   <LastModifyTime>2019-10-10 00:00:00
   <OrderId>12345</OrderId>
   <CommitterId>1</CommitterId>
   <PluginType>DC COMMON</PluginType>
   <StatusCode>success</StatusCode>
   <RelatedUserNickList>
       <UserNicks>a,b,c</UserNicks>
   </RelatedUserNickList>
   <StatusDesc>success</StatusDesc>
   <WorkflowStatusDesc>approved</WorkflowStatusDesc>
   <WorkflowInstanceId>1</WorkflowInstanceId>
   <Committer>xxx</Committer>
   <RelatedUserList>
       <UserIds>1,2,3</UserIds>
   </RelatedUserList>
</OrderBaseInfo>
<RequestId>7133DF67-5B25-460F-8285-C4CC93472C2F/RequestId>
<ErrorCode>UnknownError
<ErrorMessage>UnknownError
<Success>true</Success>
```

JSON 格式

{"OrderBaseInfo":{"Comment":"test","CreateTime":"2019-10-10 00:00:00","LastModifyTime":"2019-10-10 00:00:00","OrderId":"12345","CommitterId":"1","PluginType":"DC_COMMON","StatusCode":"success","RelatedUserNickList":{"UserNicks":"a,b,c"},"StatusDesc":"success","WorkflowStatusDesc":"approved","WorkflowInstanceId":"1","Committer":"xxxx","RelatedUserList":{"UserIds":"1,2,3"}},"RequestId":"7133DF67-5B25-460F-8285-C4CC93472C2F","ErrorCode":"UnknownError","ErrorMessage":"UnknownError","Success":"true"}

6.7.1.5. ListOrders

调用ListOrders接口获取工单列表信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	List Orders	系统规定参数。取值:List Orders。
Tid	Long	否	-1	租户ID。

名称	类型	是否必选	示例值	描述
PluginType	String	否	DC_COMMON	工单类型,取值和说明如下: PERM_APPLY: 权限申请 OWNER_APPLY: owner申请 DATA_CORRECT: 数据变更 NDDL: 结构设计 DATA_TRACK: 数据追踪 TABLE_SYNC: 库表同步 DATA_EXPORT: 数据导出 SENSITIVITY: 敏感列等级调整
OrderResultType	String	否	AS_ADMIN	搜索类型,取值和说明如下: AS_ADMIN: 工单大盘 AS_COMMITTER: 我发起的工单 AS_HANDLER: 待我处理的工单 AS_OWNER: 我已处理的工单 AS_Related: 与我相关
SearchDateType	String	否	CREATE_TIME	时间搜索条件,取值: ◆ CREATE_TIME: 创建时间 ◆ MODIFY_TIME: 修改时间
StartTime	String	否	2019-10-10 00:00:00	开始时间。
EndTime	String	否	2019-10-10 00:00:00	结束时间。
SearchContent	String	否	test	搜索内容。
OrderStatus	String	否	ALL	工单状态,取值和说明如下: ◆ ALL: 不限 ◆ FINISHED: 已结束 ◆ RUNNING: 进行中
PageSize	Integer	否	10	每页记录数。
PageNumber	Integer	否	1	第几页。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
Orders	Array of Order		工单列表信息。
Order			
Comment	String	test	工单备注信息。
Committer	String	test	工单提交者。
CommitterId	Long	1	工单提交者ID。
CreateTime	String	2019-10-10 00:00:00	工单创建时间。
LastModifyTim e	String	2019-10-10 00:00:00	工单最后修改时间。
Orderld	Long	1432	工单ID。
PluginType	String	DC_COMMON	工单类型。
StatusCode	String	success	工单状态码。
StatusDesc	String	success	工单状态描述。
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。
Success	Boolean	true	请求是否成功。
TotalCount	Long	100	记录总数。

示例

请求示例

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

正常返回示例

XML 格式

```
<TotalCount>5</TotalCount>
<RequestId>7AC17A86-9E6F-484B-AE8D-F9416986A108/RequestId>
<Orders>
   <Order>
       <StatusCode>success</StatusCode>
       <PluginType>DC COMMON</PluginType>
       <Committer>xxx[子账号]</Committer>
       <StatusDesc>数据变更成功</StatusDesc>
       <LastModifyTime>2019-11-19 13:53:05/LastModifyTime>
       <CreateTime>2019-11-19 13:52:36
       <OrderId>737160</OrderId>
       <Comment>测试数据变更</Comment>
       <CommitterId>104442/CommitterId>
   </Order>
   <Order>
       <StatusCode>new</StatusCode>
       <PluginType>DC COMMON</PluginType>
       <Committer>xxx[子账号]</Committer>
       <StatusDesc> 待提交审批</StatusDesc>
       <LastModifyTime>2019-10-29 20:51:37</LastModifyTime>
       <CreateTime>2019-10-29 20:51:36</CreateTime>
       <OrderId>737116</OrderId>
       <Comment>测试创建数据变更工单</Comment>
       <CommitterId>104442//CommitterId>
   </Order>
   <Order>
       <StatusCode>closed</StatusCode>
       <PluginType>DC COMMON</PluginType>
       <Committer>xxx[子账号]</Committer>
       <StatusDesc>工单已关闭: 还是不支持库级别操作</StatusDesc>
       <LastModifyTime>2019-10-20 22:09:38</LastModifyTime>
       <CreateTime>2019-10-20 22:08:19</CreateTime>
       <OrderId>737089</OrderId>
       <Comment>测试建库</Comment>
       <CommitterId>104442</CommitterId>
   </Order>
   <Order>
       <StatusCode>new</StatusCode>
       <PluginType>DC COMMON</PluginType>
       <Committer>xxx[子账号]</Committer>
       <StatusDesc>预检查失败</StatusDesc>
       <LastModifyTime>2019-09-30 16:58:35</LastModifyTime>
       <CreateTime>2019-09-30 16:58:34
       <OrderId>737016</OrderId>
       <Comment>测试</Comment>
       <CommitterId>104442//CommitterId>
   </Order>
```

JSON 格式

```
"TotalCount": 5,
"RequestId": "7AC17A86-9E6F-484B-AE8D-F9416986A108",
"Orders": {
  "Order": [
     "StatusCode": "success",
     "PluginType": "DC COMMON",
      "Committer": "xxx[子账号]",
     "StatusDesc": "数据变更成功",
     "LastModifyTime": "2019-11-19 13:53:05",
     "CreateTime": "2019-11-19 13:52:36",
      "OrderId": 737160,
     "Comment": "测试数据变更",
     "CommitterId": 104442
    },
     "StatusCode": "new",
     "PluginType": "DC COMMON",
     "Committer": "xxx[子账号]",
     "StatusDesc": "待提交审批",
     "LastModifyTime": "2019-10-29 20:51:37",
     "CreateTime": "2019-10-29 20:51:36",
     "OrderId": 737116,
      "Comment": "测试创建数据变更工单",
     "CommitterId": 104442
    },
     "StatusCode": "closed",
     "PluginType": "DC_COMMON",
     "Committer": "xxx[子账号]",
     "StatusDesc": "工单已关闭: 还是不支持库级别操作",
      "LastModifyTime": "2019-10-20 22:09:38",
     "CreateTime": "2019-10-20 22:08:19",
     "OrderId": 737089,
     "Comment": "测试建库",
      "CommitterId": 104442
```

 数据管理 开发指南·API参考

```
"StatusCode": "new",
      "PluginType": "DC_COMMON",
      "Committer": "xxx[子账号]",
      "StatusDesc": "预检查失败",
      "LastModifyTime": "2019-09-30 16:58:35",
      "CreateTime": "2019-09-30 16:58:34",
      "OrderId": 737016,
      "Comment": "测试",
      "CommitterId": 104442
      "StatusCode": "success",
     "PluginType": "DC_COMMON",
      "Committer": "xxx[子账号]",
     "StatusDesc": "数据变更成功",
      "LastModifyTime": "2019-08-17 14:07:54",
      "CreateTime": "2019-08-17 14:07:42",
      "OrderId": 736774,
      "Comment": "测试普通数据变更",
      "CommitterId": 104442
 ]
},
"Success": true
```

6.7.1.6. SubmitOrderApproval

调用Submit OrderApproval接口提交工单审批。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	Submit OrderAppr oval	系统规定参数。取 值:SubmitOrderApproval。
Orderld	Long	是	12345	工单ID。
Tid	Long	是	-1	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=SubmitOrderApproval
&OrderId=12345
&Tid=-1
&<公共请求参数>
```

正常返回示例

XML 格式

<RequestId>7AC17A86-9E6F-484B-AE8D-F9416986A108</RequestId>
<Success>true</Success>

JSON 格式

```
{
    "RequestId": "7AC17A86-9E6F-484B-AE8D-F9416986A108",
    "Success": true
}
```

6.7.1.7. CreateOrder

调用CreateOrder接口创建工单。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateOrder	系统规定参数。取值:CreateOrder。
Comment	String	是	test	创建工单描述信息。

名称	类型	是否必选	示例值	描述
PluginParam	Json	是	0	创建工单参数,格式为JSON形式的字符 串,每个工单类型的参数设置不一致,请 参考补充说明。
PluginType	String	是	DAT A_EXPORT	工单类型,目前支持的工单类型,取值和说明如下: DC_COMMON:普通数据变更 DC_CHUNK:无锁数据变更 DC_CRON_CLEAR:历史数据清理 NDDL:结构设计 DATA_EXPORT:数据导出 PERM_APPLY: 权限申请 OWNER_APPLY: Owner申请
RelatedUserList	String	是	1,2,3	工单相关人ID列表,多个相关人英文半角 逗号(,)隔开。
Tid	Long	否	-1	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

PluginParam参数补充说明

```
json
# 普通数据变更
"pluginParam" : {
   "classify" : "测试", // 原因分类
   "estimateAffectRows" : 1, // 用户预估影响行数
   "execSQL" : "update base_op_log set id=1 where id=1;", // 执行SQL
   "dbItemList": [ { // 执行的db列表
     "dbId" : 179694,
     "logic" : false
   } ],
   "rbSQL": "update base op log set id=1 where id=1;", // 用户填写的回滚SQL
   "rbAttachmentName" : null, // 回滚SQL附件
   "sqlType" : "TEXT", // SQL类型,取值为TEXT, ATTACHMENT
   "rbSQLType" : "TEXT", // 回滚SQL类型,取值同上
   "attachmentName" : // SQL附件
# 无锁数据变更
"pluginParam" : {
   "classify" : "测试",
   "execSQL" : "update base_op_log set op_type=op_type where 1=1;", // 执行SQL
   "dbItemList" : [ {
     "dbId" : 179694, // 变更库id
     "logic" : false
   } ]
# 历史数据清理
"pluginParam" : {
   "duration": 10, // 指定时长(小时), 当不指定时长时,值为0;
   "specifyDuration": true, // 是否指定时长, true: 是, false: 否,默认是false
   "classify" : "测试",
   "cronFormat": "0 1 21 * * ?", // 定时执行的表达式
   "execSql" : null,
   "dbItemList" : [ {
     "dbId" : 179694, // 变更库id
     "logic" : false
   "cronClearItemList" : [ {
     "filterSQL" : null, // 过滤条件
     "columnName" : "gmt create", // 时间字段
     "remainDays" : 60, // 保留时长
     "tableName" : "base app" // 表名
   } ]
```

```
json
# 结构设计
"pluginParam": {
    "title": "test", // 标题
    "description": "test", // 描述
    "dbId": 118989, // 基准库
    "logic": false, // 是否逻辑库
    "relatedIds": [], // 变更相关人id
}
```

```
json
# 数据导出
"pluginParam": {
  "classify": "原因分类", // 原因分类
  "dbId": 179719, // 导出库id
  "exeSQL": "select 1", // 导出SQL
  "logic": false, // 是否逻辑库
  "ignoreAffectRows": false, // 是否忽略校验影响行数
  "affectRows": 1, // 用户预估影响行数
  "ignoreAffectRowsReason": "" // 忽略影响行数校验原因
}
```

返回数据

名称	类型	示例值	描述
CreateOrderRes ult	List	123	创建工单成功后返回的工单号。
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CreateOrder
&Comment=test
&PluginParam={}
&PluginType=DATA_EXPORT
&RelatedUserList=1,2,3
&<公共请求参数>
```

正常返回示例

XML 格式

JSON 格式

```
"RequestId": "427688B8-ADFB-4C4E-9D45-EF5C1FD6E23D",

"CreateOrderResult": {
    "OrderIds": [123]
},
    "Success": true
}
```

6.7.2. 数据变更

6.7.2.1. ExecuteDataCorrect

调用ExecuteDataCorrect接口执行数据变更工单。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ExecuteDataCorre ct	系统规定参数。取 值:ExecuteDataCorrect。
Orderld	Long	是	123	工单ID。
Tid	String	否	-1	租户ID。

数据管理 开发指南·API参考

名称	类型	是否必选	示例值	描述
ActionDetail	Json	否	{ "startTime" : null, "endTime" : null, "transaction" : false, "backupData" : true }	执行数据变更的参数设置: json "actionDetail":{ "startTime": null, // 设置定时执行的执行时间,若您选择立即执行不需要设置该参数。 "endTime": null, // 设置定时执行的结束时间,若您选择立即执行不需要设置该参数。 "transaction": false, // 选择是否以事务方式运行。 "backupData": true // 选择是否备份数据。 }
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=ExecuteDataCorrect
&OrderId=123
&<公共请求参数>
```

正常返回示例

XML 格式

< RequestId > 7133DF67 - 5B25 - 460F - 8285 - C4CC93472C2F < / RequestId > < Success > true < / Success >

JSON 格式

```
{
    "RequestId": "7133DF67-5B25-460F-8285-C4CC93472C2F",
    "Success": true
}
```

6.7.2.2. GetDataCorrectBackupFiles

调用Get Dat aCorrect Backup Files接口获取数据变更备份文件附件下载地址。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	Get Dat a Correct Ba ckup Files	系统规定参数。取 值:GetDataCorrectBackupFiles。
Orderld	Long	是	12345	工单ID。
Tid	Long	否	-1	租户ID。
ActionDetail	Json	否	0	操作参数,无需传递。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
DataCorrectBac kupFiles	List	XXX	备份文件附件下载地址信息。
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。

名称	类型	示例值	描述
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetDataCorrectBackupFiles
&OrderId=12345
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DataCorrectBackupFiles>
        <FileUrl>https://xxx.oss-cn-hangzhou.aliyuncs.com/xxx.zip?Expires=1574382979&amp;OSSAcc
essKeyId=wSa97B52GzF4fGqu&amp;Signature=ykWSysajKcQjXjG0X4ApOQv1IXQ%3D</FileUrl>

<
```

JSON 格式

6.7.2.3. GetDataCorrectOrderDetail

调用Get Dat a Correct OrderDet ail接口获取数据变更工单详情信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetDataCorrectOr derDetail	系统规定参数。取 值:GetDataCorrectOrderDetail。
Orderld	Long	是	12345	工单ID。

名称	类型	是否必选	示例值	描述
Tid	Long	否	-1	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
DataCorrectOrd erDetail	Struct		数据变更工单详情信息。
DatabaseList	Array of Database		变更数据库信息。
Database			
Dbld	Integer	123456	数据库ID。
DbType	String	mysql	数据库类型。
EnvType	String	product	环境类型。
Logic	Boolean	false	是否为逻辑库。
SearchName	String	xxx@xxx:3306	逻辑库的搜索名。
OrderDetail	Struct		工单详情信息。
ActualAffectRo ws	Long	100	系统校验的影响行数。
AttachmentNa me	String	XXX	SQL附件。
Classify	String	test	变更原因分类。
EstimateAffect Rows	Long	100	用户预估影响行数。
ExeSQL	String	update t1 set name = 'xxx' where id <= 100	执行的SQL。

名称	类型	示例值	描述
IgnoreAffectRo ws	Boolean	false	是否忽略预校验。
IgnoreAffectRo wsReason	String	test	忽略预校验原因。
RbAttachment Name	String	test	回滚SQL附件。
RbSQL	String	empty	用户填写的回滚SQL。
RbSQLType	String	text	回滚SQL类型,取值和说明如下: TEXT: 文本 ATTACHMENT: 附件
SqlType	String	text	SQL类型,取值和说明如下: ● TEXT:文本 ● ATTACHMENT:附件
PreCheckDetail	Array of TaskCheckDO		预校验详情信息。
TaskCheckDO			
CheckStatus	String	SUCCESS	预校验状态,取值和说明如下: WAITING: 等待检查 RUNNING: 检查中 SUCCESS: 通过 FAIL: 不通过
CheckStep	String	PERMISSION_CHECK	检查步骤,取值和说明如下: SQL_PARSE:解析 SQL_TYPE_CHECK:类型检查 PERMISSION_CHECK:权限检查 ROW_CHECK:行数检查
UserTip	String	tip messsage	校验步骤提示信息。
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。

名称	类型	示例值	描述
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetDataCorrectOrderDetail
&OrderId=12345
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>ACE263E5-CF5F-44DD-A7B1-42C5E78E9650/RequestId>
<DataCorrectOrderDetail>
   <PreCheckDetail>
       <TaskCheckDO>
           <CheckStatus>SUCCESS</CheckStatus>
           <CheckStep>SQL PARSE</CheckStep>
       </TaskCheckDO>
       <TaskCheckDO>
           <CheckStatus>SUCCESS</CheckStatus>
           <CheckStep>SQL TYPE CHECK</CheckStep>
       </TaskCheckDO>
       <TaskCheckDO>
           <CheckStatus>SUCCESS</CheckStatus>
           <CheckStep>PERMISSION CHECK</CheckStep>
       </TaskCheckDO>
       <TaskCheckDO>
           <CheckStatus>SUCCESS</CheckStatus>
           <userTip>系统统计值,实际影响行数仍然以SQL执行为准!</userTip>
           <CheckStep>ROW_CHECK</CheckStep>
       </TaskCheckDO>
   </PreCheckDetail>
   <OrderDetail>
       <ActualAffectRows>0</ActualAffectRows>
       <RbSQLType>TEXT
       <ExeSQL>update base_user set id = 1 where id = 1;</ExeSQL>
       <EstimateAffectRows>1</EstimateAffectRows>
       <SqlType>TEXT</SqlType>
       <Classify>test</Classify>
       <IgnoreAffectRowsReason/>
   </OrderDetail>
</DataCorrectOrderDetail>
<Success>true</Success>
```

JSON 格式

```
"RequestId": "ACE263E5-CF5F-44DD-A7B1-42C5E78E9650",
"DataCorrectOrderDetail": {
 "PreCheckDetail": {
   "TaskCheckDO": [
        "CheckStatus": "SUCCESS",
       "CheckStep": "SQL PARSE"
      },
       "CheckStatus": "SUCCESS",
       "CheckStep": "SQL TYPE CHECK"
        "CheckStatus": "SUCCESS",
       "CheckStep": "PERMISSION_CHECK"
      },
        "CheckStatus": "SUCCESS",
       "UserTip": "系统统计值,实际影响行数仍然以SQL执行为准!",
       "CheckStep": "ROW CHECK"
   ]
  },
  "OrderDetail": {
   "ActualAffectRows": 0,
    "RbSQLType": "TEXT",
   "ExeSQL": "update base_user set id = 1 where id = 1;",
   "EstimateAffectRows": 1,
    "SqlType": "TEXT",
    "Classify": "test",
   "IgnoreAffectRowsReason": ""
},
"Success": true
```

6.7.3. 结构设计

6.7.3.1. CreatePublishGroupTask

调用CreatePublishGroupTask接口执行结构设计变更。

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	CreatePublishGro upTask	系统规定参数。取 值:CreatePublishGroupTask。
Dbld	Integer	是	12345	执行的数据库ID。
Logic	Boolean	是	false	是否为逻辑库。
Orderld	Long	是	142435	工单ID。
PublishStrategy	String	是	IMMEDIATELY	执行策略,取值和说明: ● IMMEDIATELY: 立即执行 ● REGULARLY: 定时执行
Tid	Long	是	-1	租户ID。
PlanTime	String	否	2019-10-10 00:00:00	定时执行计划时间。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。
Success	Boolean	true	请求是否成功。
Taskld	Long	413452	任务ID。

示例

请求示例

正常返回示例

XML 格式

```
<RequestId>78C93646-D9B1-4E6E-B5FD-C69F94A7905A</RequestId>
<Success>true</Success>
<TaskId>13424</TaskId>
```

JSON 格式

```
{
   "RequestId": "78C93646-D9B1-4E6E-B5FD-C69F94A7905A",
   "Success": true,
   "TaskId": 13424
}
```

6.7.4. 数据导出

6.7.4.1. ExecuteDataExport

调用ExecuteDataExport接口执行数据导出工单。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ExecuteDataExpo rt	系统规定参数。取 值:ExecuteDataExport。
ActionDetail	Json	是	{ "fileType": "CSV", "encoding": "" }	执行操作参数: json { "fileType": "CSV", // 导出 格式 "encoding": "" // 编码 }

名称	类型	是否必选	示例值	描述
Orderld	Long	是	1234	工单ID。
Tid	Long	是	-1	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	FE8EE2F1-4880- 46BC-A704- 5CF63EAF9A04	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?ActionDetail={ "fileType": "CSV", "encoding": "" }
&OrderId=1234
&Tid=-1
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>FE8EE2F1-4880-46BC-A704-5CF63EAF9A04</RequestId>
<Success>true</Success>
```

JSON 格式

```
{
   "RequestId": "FE8EE2F1-4880-46BC-A704-5CF63EAF9A04",
   "Success": true
}
```

6.7.4.2. GetDataExportDownloadURL

调用Get Dat aExport Download URL接口获取数据导出结果附件下载地址。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	Get Dat a Export Do wnload URL	系统规定参数。取 值:GetDataExportDownloadURL。
Orderld	Long	是	12345	工单ID。
Tid	Long	是	-1	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
DownloadURLR esult	Struct		导出文件下载地址。
HasResult	Boolean	true	是否有下载结果。
TipMessage	String	tip message	异常提示信息。
URL	String	xxx	导出文件下载地址。
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetDataExportDownloadURL
&OrderId=12345
&Tid=-1
&<公共请求参数>
```

正常返回示例

XML 格式

JSON 格式

```
{
  "DownloadURLResult": {
    "URL": "https://dms-idb-hangzhou.oss-cn-hangzhou.aliyuncs.com/xxx.zip?Expires=157438431
3&OSSAccessKeyId=wSa97B52GzF4fGqu&Signature=HQQWBRSKiVV6cupugh1/eAVSqXg%3D",
    "HasResult": true
},
    "RequestId": "4BF24EA5-9013-4C85-AE68-6C23AF5E0097",
    "Success": true
}
```

6.7.4.3. GetDataExportOrderDetail

调用Get DataExport OrderDetail接口获取数据导出工单详情。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetDataExportOr derDetail	系统规定参数。取 值:GetDataExportOrderDetail。
Orderld	Long	是	123456	工单ID。
Tid	Long	是	-1	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
DataExportOrd erDetail	Struct		数据导出工单详情信息。
KeyInfo	Struct		状态信息。
JobStatus	String	EXPORT_SUCCESS	导出工单状态,取值和说明如下: PRE_CHECKING: 执行预校验 PRE_CHECK_SUCCESS: 预校验成功 PRE_CHECK_FAIL: 预校验失败 WAITING_APPLY_AUDIT: 等待提交审批 APPLY_AUDIT_SUCESS: 提交审批成功 ENABLE_EXPORT: 审批通过,允许导出 WAITING_EXPORT: 转待调度进行导出 DOING_EXPORT: 执行导出 EXPORT_FAIL: 导出失败 EXPORT_SUCCESS: 导出成功
PreCheckId	Long	123	预校验ID。
OrderDetail	Struct		工单详情信息。
ActualAffectRo ws	Long	1	系统校验影响行数。
Classify	String	text	导出原因分类。
Database	String	xxx@xxx:3306	导出数据库。
Dbld	Integer	123	导出数据库ID。
EnvType	String	test	导出数据库环境信息。
ExeSQL	String	select 1	执行的导出SQL。
IgnoreAffectRo ws	Boolean	false	是否忽略影响行数。
lgnoreAffectRo wsReason	String	empty	忽略校验原因。

名称	类型	示例值	描述
Logic	Boolean	false	是否逻辑库。
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	427688B8-ADFB- 4C4E-9D45- EF5C1FD6E23D	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetDataExportOrderDetail
&OrderId=123456
&Tid=-1
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DataExportOrderDetail>
   <KeyInfo>
       <PreCheckId>15544</PreCheckId>
       <JobStatus>EXPORT_SUCCESS</JobStatus>
   </KeyInfo>
   <OrderDetail>
       <Logic>false</Logic>
       <ActualAffectRows>1</ActualAffectRows>
       <ExeSQL>select 1;</ExeSQL>
       <Database>yuyang_test@xxx:3306 [yuyang_test_xxx] 杭州 非单元化
       <IgnoreAffectRows>false</IgnoreAffectRows>
       <DbId>116367</pbId>
       <Classify>测试</Classify>
       <EnvType>product</EnvType>
   </OrderDetail>
</DataExportOrderDetail>
<RequestId>78C93646-D9B1-4E6E-B5FD-C69F94A7905A/RequestId>
<Success>true</Success>
```

JSON 格式

```
"DataExportOrderDetail": {
    "KeyInfo": {
        "PreCheckId": 15544,
        "JobStatus": "EXFORT_SUCCESS"
    },
    "OrderDetail": {
        "Logic": false,
        "ActualAffectRows": 1,
        "ExeSQL": "select 1;",
        "Database": "yuyang_test@xxx:3306 [yuyang_test_xxx] 杭州 非单元化",
        "IgnoreAffectRows": false,
        "DbId": 116367,
        "Classify": "测试",
        "EnvType": "product"
    }
},
    "RequestId": "78C93646-D9B1-4E6E-B5FD-C69F94A7905A",
    "Success": true
}
```

6.8. 权限管理

6.8.1. RevokeUserPermission

调用RevokeUserPermission接口回收用户权限。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	RevokeUserPermi ssion	系统规定参数,取 值:RevokeUserPermission。
Dbld	String	是	142345	数据库ID。
DsType	String	是	DAT ABASE	资源类型,取值和说明请参 考GrantUserPermission接口。
PermTypes	String	是	QUERY	权限类型,取值和说明请参 考GrantUserPermission接口。
Tid	Long	是	-1	租户ID。
UserAccessId	String	是	123	用户权限信息ID。

名称	类型	是否必选	示例值	描述
				用户ID,对应 GetUser 里面的用户ID。
Userid	String	是	1234	② 说明 注意不是用户的阿里云 UID。
Logic	Boolean	否	false	是否为逻辑库,取值: • true • false
TableId	String	否	1342345	表ID,授权表权限时需要指定。
TableName	String	否	table_name	表名,授权表权限时需要指定。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误消息。
RequestId	String	A99CD576-1E18- 4E86-931E- C3CCE56DC030	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

正常返回示例

XML 格式

< RequestId > A99CD576-1E18-4E86-931E-C3CCE56DC030 < / RequestId > < Success > true < / Success >

JSON 格式

```
{
   "RequestId": "A99CD576-1E18-4E86-931E-C3CCE56DC030",
   "Success": true
}
```

6.8.2. ListDatabaseUserPermssions

调用List Dat abaseUserPermssions接口获取具备指定数据库的用户权限详情。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	List Dat abase User Permssions	要执行的操作 <i>,</i> 取 值:List Dat abaseUserPermssions。
Dbld	String	是	123	数据库ID。
PermType	String	是	DAT ABASE	权限类型,取值和说明如下: ● DATABASE: 库权限 ● TABLE: 表权限 ● COLUMN: 列权限
Tid	Long	是	-1	租户ID。
Logic	Boolean	否	false	是否逻辑库。
UserName	String	否	search_user_nam e	用户昵称。
PageNumber	Integer	否	1	第几页。
PageSize	Integer	否	10	每页数据量。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	68075D06-7406- 4887-83A7- F558A4D28C13	请求ID。
Success	Boolean	true	请求是否成功。
TotalCount	Long	1	满足条件的记录总数。
UserPermission s	Array of UserPermission		用户权限详情。
UserPermission			
Alias	String	instance_alias	实例别名。
ColumnName	String	column_name	字段名。
Dbld	String	1234	数据库ID。
DbType	String	mysql	数据库类型。
DsType	String	DATABASE	资源权限类型。
EnvType	String	product	数据库所属环境。
InstanceId	String	1443	实例ID。
Logic	Boolean	false	是否逻辑库。
PermDet ails	Array of PermDetail		用户权限详情。
PermDetail			
CreateDate	String	2019-12-12 00:00:00	权限创建时间。

名称	类型	示例值	描述
ExpireDate	String	2020-12-12 00:00:00	权限过期时间。
ExtraData	String	DEFAULT	附加信息,预留属性。
OriginFrom	String	xxx grant	权限申请来源描述。
PermType	String	QUERY	权限类型,取值和说明如下: ● QUERY: 查询 ● EXPORT: 导出 ● CORRECT: 变更
UserAccessId	String	13434	权限记录ID。
SchemaName	String	xxx@xxxx:3306	Schema名称。
SearchName	String	XXX	数据库搜索名。
TableId	String	42345	表ID。
TableName	String	table_name	表名。
Userld	String	14324	用户ID。
UserNickName	String	user_nick_name	用户名称。

示例

请求示例

 $\label{lem:listDatabaseUserPermssions} $$ $$ \triangle ListDatabaseUserPermssions $$ \DbId=123$$

&PermType=DATABASE

&Tid=-1

&<公共请求参数>

正常返回示例

XML 格式

```
<UserPermissions>
   <UserPermission>
       <Logic>false</Logic>
       <UserNickName>xxx</UserNickName>
       <DsType>DATABASE</psType>
       <Alias>beihe_ent_prod</Alias>
       <PermDetails>
           <PermDetail>
               <ExpireDate>2020-10-31 00:00:00/ExpireDate>
               <UserAccessId>390915</UserAccessId>
               <PermType>QUERY</PermType>
               <OriginFrom>xxx</OriginFrom>
               <CreateDate>2019-11-21 15:07:45
           </PermDetail>
           <PermDetail>
               <ExpireDate>2020-10-31 00:00:00</ExpireDate>
               <UserAccessId>390916</UserAccessId>
               <PermType>EXPORT</PermType>
               <OriginFrom>xxx</OriginFrom>
               <CreateDate>2019-11-21 15:07:45
           </PermDetail>
           <PermDetail>
               <ExpireDate>2020-10-31 00:00:00/ExpireDate>
               <UserAccessId>390917</UserAccessId>
               <PermType>CORRECT</PermType>
               <OriginFrom>xxx</OriginFrom>
               <CreateDate>2019-11-21 15:07:45
           </PermDetail>
       </PermDetails>
       <InstanceId>1434</InstanceId>
       <SearchName>xxx@xxxx:3306</SearchName>
       <DbId>123434</pbId>
       <UserId>432423
       <DbType>drds</DbType>
       <EnvType>product</EnvType>
       <SchemaName>xxx</SchemaName>
    </UserPermission>
</UserPermissions>
<TotalCount>1</TotalCount>
<RequestId>68075D06-7406-4887-83A7-F558A4D28C13/RequestId>
<Success>true</Success>
```

JSON 格式

```
"UserPermissions": {
  "UserPermission": [
      "Logic": false,
      "UserNickName": "xxx",
      "DsType": "DATABASE",
      "Alias": "beihe_ent_prod",
      "PermDetails": {
        "PermDetail": [
            "ExpireDate": "2020-10-31 00:00:00",
            "UserAccessId": "390915",
            "PermType": "QUERY",
            "OriginFrom": "xxx",
            "CreateDate": "2019-11-21 15:07:45"
          },
            "ExpireDate": "2020-10-31 00:00:00",
            "UserAccessId": "390916",
            "PermType": "EXPORT",
            "OriginFrom": "xxx",
            "CreateDate": "2019-11-21 15:07:45"
          },
            "ExpireDate": "2020-10-31 00:00:00",
            "UserAccessId": "390917",
            "PermType": "CORRECT",
            "OriginFrom": "xxx",
            "CreateDate": "2019-11-21 15:07:45"
       ]
      "InstanceId": 1434,
      "SearchName": "xxx@xxxx:3306",
      "DbId": 123434,
      "UserId": 432423,
      "DbType": "drds",
      "EnvType": "product",
      "SchemaName": "xxx"
 ]
},
"TotalCount": 1,
"RequestId": "68075D06-7406-4887-83A7-F558A4D28C13",
"Success": true
```

6.8.3. SetOwners

调用SetOwners接口设置库表Owner。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	Set Owners	系统规定参数,取值:Set Owners。
Ownerlds	String	是	123, 456	设置为Owner的用户ID,多个用户ID用逗号分开。
OwnerType	String	是	DAT ABASE	Owner类型,取值和说明如下: DATABASE: 物理库Owner LOGIC_DATABASE: 逻辑库Owner TABLE: 物理表Owner LOGIC_TABLE: 逻辑表Owner
Resourceld	String	是	123	资源ID,根据OwnerType的不同,指向不同的资源,OwnerType和对应资源ID的解释如下: DATABASE:物理库ID LOGIC_DATABASE:逻辑库ID TABLE:物理表ID LOGIC_TABLE:逻辑表ID
Tid	Long	是	-1	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误消息。
RequestId	String	A99CD576-1E18- 4E86-931E- C3CCE56DC030	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=SetOwners
&OwnerIds=123, 456
&OwnerType=DATABASE
&ResourceId=123
&Tid=-1
&<公共请求参数>
```

正常返回示例

XML 格式

<RequestId>A99CD576-1E18-4E86-931E-C3CCE56DC030</RequestId><Success>true</Success>

JSON 格式

```
{
   "RequestId": "A99CD576-1E18-4E86-931E-C3CCE56DC030",
   "Success": true
}
```

6.8.4. ListUserPermissions

调用ListUserPermissions接口获取指定用户的库表权限。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListUserPermissio ns	要执行的操作,取 值:ListUserPermissions。
PermType	String	是	DAT ABASE	资源权限类型,取值和说明请参 考List Dat abase User Permssions。
Tid	Long	是	-1	租户ID。
Userld	String	是	13435	用户ID,对应 GetUser 里面的用户ID,注 意不是用户的阿里云UID。
DatabaseName	String	否	db_name	数据库名。

名称	类型	是否必选	示例值	描述
Logic	Boolean	否	false	是否逻辑库,取值: • true • false
EnvType	String	否	dev	数据库所属环境类型。
DbType	String	否	MySQL	数据库类型。
PageNumber	Integer	否	1	页码。
PageSize	Integer	否	1	每页记录数。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	C51420E3-144A- 4A94-B473- 8662FCF4AD10	请求ID。
Success	Boolean	true	请求是否成功。
TotalCount	Long	6	满足条件记录总数。
UserPermission s	Array of UserPermission		用户权限信息列表。
UserPermission			
Alias	String	instance_alias	实例别名。
ColumnName	String	column_name	字段名。

名称	类型	示例值	描述
Dbld	String	1324	数据库ID。
DbType	String	mysql	数据库类型。
DsType	String	DAT ABASE	权限资源类型。
EnvType	String	dev	数据库所属环境类型。
InstanceId	String	134	实例ID。
Logic	Boolean	false	是否逻辑库,取值: • true • false
PermDetails	Array of PermDetail		权限详情。
PermDet ail			
CreateDate	String	2019-12-12 00:00:00	权限授予时间
ExpireDate	String	2020-12-12 00:00:00	授权到期时间
ExtraData	String	XXX	额外数据,预留参数
OriginFrom	String	XXX	权限授予来源信息
PermType	String	QUERY	权限类型
UserAccessId	String	1345	授权记录ID
SchemaName	String	test_db	数据库名。
SearchName	String	test_db@xxx:3306	库搜索名。
TableId	String	13434	表ID。

名称	类型	示例值	描述
TableName	String	test_table	表名。
Userld	String	423245	用户ID。
UserNickName	String	nick_name	用户名。

示例

请求示例

http(s)://[Endpoint]/?Action=ListUserPermissions
&PermType=DATABASE

&Tid=-1

4114 1

&UserId=13435

&<公共请求参数>

正常返回示例

XML 格式

数据管理 开发指南·API参考

```
<UserPermissions>
   <UserPermission>
       <Logic>false</Logic>
       <UserNickName>xxx</UserNickName>
       <DsType>DATABASE
       <Alias>xxx_ent_dev</Alias>
       <PermDetails>
           <PermDetail>
               <ExpireDate>2020-10-31 00:00:00/ExpireDate>
               <UserAccessId>390913</UserAccessId>
               <PermType>EXPORT</PermType>
               <OriginFrom>xxx</OriginFrom>
               <CreateDate>2019-11-21 15:07:45</CreateDate>
           </PermDetail>
           <PermDetail>
               <ExpireDate>2020-10-31 00:00:00</ExpireDate>
               <UserAccessId>390914</UserAccessId>
               <PermType>CORRECT</PermType>
               <OriginFrom>xxx</OriginFrom>
               <CreateDate>2019-11-21 15:07:45
            </PermDetail>
       </PermDetails>
       <InstanceId>24671</InstanceId>
       <SearchName>xxx@xxx:3306</SearchName>
       <DbId>111369</DbId>
       <UserId>88658</UserId>
       <DbType>drds</DbType>
       <EnvType>dev</EnvType>
       <SchemaName>xxx ent dev</SchemaName>
    </UserPermission>
</UserPermissions>
<TotalCount>6</TotalCount>
<RequestId>C51420E3-144A-4A94-B473-8662FCF4AD10/RequestId>
<Success>true</Success>
```

JSON 格式

```
"UserPermissions": {
  "UserPermission": [
      "Logic": false,
      "UserNickName": "xxx",
      "DsType": "DATABASE",
      "Alias": "xxx_ent_dev",
      "PermDetails": {
        "PermDetail": [
            "ExpireDate": "2020-10-31 00:00:00",
            "UserAccessId": "390913",
            "PermType": "EXPORT",
            "OriginFrom": "xxx",
            "CreateDate": "2019-11-21 15:07:45"
          },
            "ExpireDate": "2020-10-31 00:00:00",
            "UserAccessId": "390914",
            "PermType": "CORRECT",
            "OriginFrom": "xxx",
            "CreateDate": "2019-11-21 15:07:45"
       ]
      },
      "InstanceId": 24671,
      "SearchName": "xxx@xxx:3306",
      "DbId": 111369,
      "UserId": 88658,
      "DbType": "drds",
      "EnvType": "dev",
      "SchemaName": "xxx ent dev"
   }
 ]
},
"TotalCount": 6,
"RequestId": "C51420E3-144A-4A94-B473-8662FCF4AD10",
"Success": true
```

6.8.5. GrantUserPermission

调用GrantUserPermission接口进行用户库表权限授予。

请求参数

名称	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	Grant User Permiss ion	系统规定参数。取 值:GrantUserPermission。
Dbld	String	是	1234	数据库ID。
DsType	String	是	DAT ABASE	权限类型,取值和说明如下: DATABASE: 物理库权限 LOGIC_DATABASE: 逻辑库权限 TABLE: 物理表权限 LOGIC_TABLE: 逻辑表权限
ExpireDate	String	是	2019-12-12 00:00:00	权限过期时间。
PermTypes	String	是	QUERY, EXPORT, CORRECT	权限类型,多个权限类型用逗号隔开,取值和说明如下: • QUERY: 查询权限 • EXPORT: 导出权限 • CORRECT: 变更权限
Tid	Long	是	-1	租户ID。
Userld	String	是	12345	用户ID,对应 Get User 接口里面的用户ID,注意不是用户的阿里云UID。
Logic	Boolean	否	false	是否逻辑库。
TableId	String	否	1324245	表ID,授权表权限时需要指定。
TableName	String	否	table_name	表名,授权表权限时需要指定。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。

名称	类型	示例值	描述
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	A99CD576-1E18- 4E86-931E- C3CCE56DC030	请求ID。
Success	Boolean	true	请求是否成功。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GrantUserPermission
&DbId=1234
&DsType=DATABASE
&ExpireDate=2019-12-12 00:00:00
&PermTypes=QUERY, EXPORT, CORRECT
&Tid=-1
&UserId=12345
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>A99CD576-1E18-4E86-931E-C3CCE56DC030</RequestId>
<Success>true</Success>
```

JSON 格式

```
{
   "RequestId": "A99CD576-1E18-4E86-931E-C3CCE56DC030",
   "Success": true
}
```

6.9. 安全管理

6.9.1. ListSensitiveColumns

调用ListSensitiveColumns接口搜索敏感字段列表。

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	ListSensitiveColu mns	系统规定参数。取 值:ListSensitiveColumns。
Tid	Long	是	-1	租户ID。
SchemaName	String	否	test	数据库名。
TableName	String	否	test	表名。
ColumnName	String	否	test	字段名。
SecurityLevel	String	否	SENSITIVE	字段敏感等级,取值和说明如下: INNER: 内部 SENSITIVE: 敏感 CONFIDENTIAL: 机密
PageNumber	Integer	否	1	页码。
PageSize	Integer	否	10	每页数据量。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	0E714EF1-E617- 4BBC-A3D2- 9552063BA942	请求ID。
SensitiveColum nList	Array of SensitiveColum n		敏感字段列表。
SensitiveColum n			

名称	类型	示例值	描述
ColumnCount	Long	1	字段个数。
ColumnName	String	test	字段名。
FunctionType	String	DEFAULT	脱敏算法类型,取值和说明参考: ◆ DEFAULT: 全遮掩(默认) ◆ FIX_POS: 固定位置 ◆ FIX_CHAR: 固定字符
SchemaName	String	test	数据库名。
SecurityLevel	String	SENSITIVE	敏感字段级别。
TableName	String	test	表名。
Success	Boolean	true	请求是否成功。
TotalCount	Long	10	数据总数。

示例

请求示例

http(s)://[Endpoint]/?Action=ListSensitiveColumns
&Tid=-1

&<公共请求参数>

正常返回示例

XML 格式

数据管理 开发指南·API参考

```
<TotalCount>5</TotalCount>
<RequestId>0E714EF1-E617-4BBC-A3D2-9552063BA942/RequestId>
<SensitiveColumnList>
   <SensitiveColumn>
       <TableName>yuyang test</TableName>
       <ColumnName>c1</ColumnName>
       <ColumnCount>0</ColumnCount>
       <SecurityLevel>SENSITIVE</SecurityLevel>
        <SchemaName>testdb</SchemaName>
   </SensitiveColumn>
    <SensitiveColumn>
       <TableName>yuyang_test_luanma</TableName>
       <ColumnName>c1</ColumnName>
       <ColumnCount>0</ColumnCount>
       <SecurityLevel>SENSITIVE</SecurityLevel>
       <SchemaName>testdb</SchemaName>
   </SensitiveColumn>
   <SensitiveColumn>
       <TableName>yuyang test</TableName>
       <ColumnName>id</ColumnName>
        <ColumnCount>0</ColumnCount>
       <SecurityLevel>SENSITIVE</SecurityLevel>
       <SchemaName>testdb</SchemaName>
   </SensitiveColumn>
    <SensitiveColumn>
       <TableName>yuyang_test_luanma</TableName>
       <ColumnName>id</ColumnName>
       <ColumnCount>0</ColumnCount>
       <SecurityLevel>SENSITIVE</SecurityLevel>
       <SchemaName>testdb</SchemaName>
   </SensitiveColumn>
   <SensitiveColumn>
       <TableName>yuyang test json</TableName>
       <ColumnName>id</ColumnName>
       <ColumnCount>0</ColumnCount>
       <SecurityLevel>CONFIDENTIAL</SecurityLevel>
       <SchemaName>test demo</SchemaName>
   </SensitiveColumn>
</SensitiveColumnList>
<Success>true</Success>
```

JSON 格式

```
"TotalCount": 5,
"RequestId": "0E714EF1-E617-4BBC-A3D2-9552063BA942",
"SensitiveColumnList": {
  "SensitiveColumn": [
      "TableName": "yuyang test",
      "ColumnName": "c1",
      "ColumnCount": 0,
      "SecurityLevel": "SENSITIVE",
      "SchemaName": "testdb"
    },
      "TableName": "yuyang_test_luanma",
      "ColumnName": "c1",
      "ColumnCount": 0,
      "SecurityLevel": "SENSITIVE",
      "SchemaName": "testdb"
    },
      "TableName": "yuyang_test",
      "ColumnName": "id",
      "ColumnCount": 0,
      "SecurityLevel": "SENSITIVE",
      "SchemaName": "testdb"
    },
      "TableName": "yuyang_test_luanma",
      "ColumnName": "id",
      "ColumnCount": 0,
      "SecurityLevel": "SENSITIVE",
      "SchemaName": "testdb"
    },
     "TableName": "yuyang test json",
      "ColumnName": "id",
      "ColumnCount": 0,
      "SecurityLevel": "CONFIDENTIAL",
      "SchemaName": "test demo"
    }
 ]
},
"Success": true
```

6.9.2. ListSensitiveColumnsDetail

调用ListSensitiveColumnsDetail接口获取敏感字段详情信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListSensitiveColu mnsDetail	系统规定参数。取 值:ListSensitiveColumnsDetail。
ColumnName	String	是	test	字段名。
SchemaName	String	是	test	数据库名。
TableName	String	是	test	表名。
Tid	Long	是	1	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	7629888F-C9FB- 4D16-A7D3- B443FE06FBD4	请求ID。
SensitiveColum nsDetailList	Array of SensitiveColum nsDetail		敏感字段详情列表。
SensitiveColum nsDetail			
ColumnDescript ion	String	test	字段描述。
ColumnName	String	test	字段名。
ColumnType	String	varchar(32)	字段类型。
Dbld	Long	124	数据库ID。

名称	类型	示例值	描述
DbType	String	mysql	数据库类型。
EnvType	String	product	数据库所属环境。
Logic	Boolean	false	是否逻辑库。
SchemaName	String	test	数据库名。
SearchName	String	test@xxx:3306	数据库搜索名。
TableName	String	test	表名。
Success	Boolean	true	请求是否成功。

示例

请求示例

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

正常返回示例

XML 格式

```
<RequestId>7629888F-C9FB-4D16-A7D3-B443FE06FBD4</RequestId>
<SensitiveColumnsDetailList>
   <SensitiveColumnsDetail>
       <TableName>king_info</TableName>
       <ColumnName>mobile</ColumnName>
        <ColumnType>varchar(64)</ColumnType>
        <SearchName>poc prod@xxx.mysql.rds.aliyuncs.com:3306 [xxx] </SearchName>
        <ColumnDescription>mobile</ColumnDescription>
        <DbId>8511149</pbId>
       <Logic>false</Logic>
        <SchemaName>poc prod</SchemaName>
        <EnvType>product</EnvType>
        <DbType>mysql</DbType>
   </SensitiveColumnsDetail>
</SensitiveColumnsDetailList>
<Success>true</Success>
```

JSON 格式

```
"RequestId": "7629888F-C9FB-4D16-A7D3-B443FE06FBD4",
"SensitiveColumnsDetailList": {
  "SensitiveColumnsDetail": [
      "TableName": "king info",
     "ColumnName": "mobile",
      "ColumnType": "varchar(64)",
      "SearchName": "poc_prod@xxx.mysql.rds.aliyuncs.com:3306 [xxx] ",
      "ColumnDescription": "mobile",
      "DbId": 8511149,
      "Logic": false,
      "SchemaName": "poc prod",
      "EnvType": "product",
      "DbType": "mysql"
 ]
},
"Success": true
```

6.10. 数据查询

6.10.1. ExecuteScript

调用ExecuteScript接口执行SQL脚本。

目前仅支持管控模式为安全协同的实例调用该API。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ExecuteScript	系统规定参数。取值:ExecuteScript。
Dbld	Integer	是	123	数据库ID。 ② 说明 您可以通 过Get Database接口返回参 数DatabaseId中获取该值。
Logic	Boolean	是	false	是否为逻辑库。
Script	String	是	select dt from report_daily	要执行的具体SQL脚本。包括DQL、DDL和DML,其中DDL和DML是否允许依赖于实例的安全管控配置。

名称	类型	是否必选	示例值	描述
Tid	Long	否	234	租户ID。
RegionId	String	否	cn-hangzhou	地域ID。

返回数据

名称	类型	示例值	描述
ErrorCode	String	UnknownError	错误码。
ErrorMessage	String	UnknownError	错误信息。
RequestId	String	FE8EE2F1-4880- 46BC-A704- 5CF63EAF9A04	请求ID。
Results	Array of Result		脚本执行返回结果列表,列表中每一项对应一个 脚本执行结果。
ColumnNames	List	["col1", "col2"]	执行查询脚本时,返回的结果集字段列表。
Message	String	UnknownError	执行失败时,返回的错误信息。
RowCount	Long	1	返回的结果记录数。
Rows	List	[{"col1": "value1", "col2":"value2"}]	执行查询脚本时,返回的结果数据。
Success	Boolean	true	单个脚本执行是否成功。
Success	Boolean	true	请求是否成功。

示例

请求示例

http(s)://[Endpoint]/?Action=ExecuteScript
&DbId=123

&Logic=false

&Script=select dt from report_daily

&<公共请求参数>

正常返回示例

XML 格式

JSON 格式

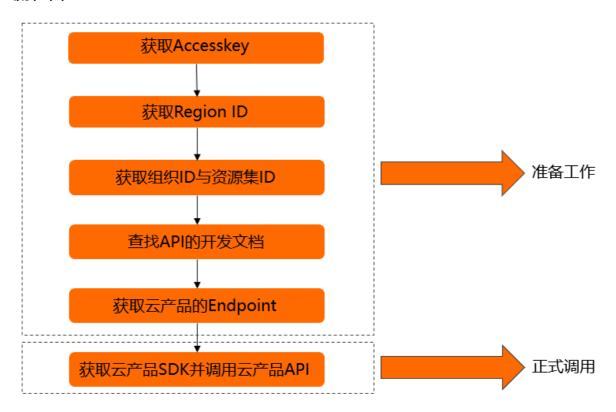
```
{"RequestId":"FE8EE2F1-4880-46BC-A704-5CF63EAF9A04","Results":[{"Message":"UnknownError","R owCount":"1","Success":"true"},{"ColumnNames":"[\"col1\", \"col2\"]","Rows":"[{\"col1\": \" value1\", \"col2\":\"value2\"}]"}],"ErrorCode":"UnknownError","ErrorMessage":"UnknownError","Success":"true"}
```

7. 附录

7.1. ASAPI网关调用流程说明

专有云中绝大多数产品都可以通过ASAPI网关进行调用,本文以通过ASAPI网关调用专有云API为例,为您介绍调用流程及可以参考的文档资料。

流程图



调用步骤

- 1. 获取ASAPI的SDK。
 - 使用ASAPI调用API前, 您需要先获取ASAPI的SDK包, 请参见<mark>获取ASAPI的SDK包</mark>。
- 2. 获取ASAPI的Endpoint,请参见获取ASAPI的Endpoint。
- 3. 获取AccessKey。

获取调用时所需要的AccessKey参数。

? 说明

推荐使用个人AccessKey调用Apsara Uni-manager运营控制台及云产品接口。

- 4. 获取RegionID。
 - 获取调用时所需要的RegionID参数。
- 5. 获取组织ID与资源集ID。

数据管理 开发指南· <mark>附录</mark>

资源集ID需要通过API进行获取,获取方式请参见*Apsara Uni-manager运营控制台开发指南*中的章节*List ResourceGroup*。

? 说明

调用API时,通过Header参数组织ID(<x-acs-organizationid>)和资源集ID(<x-acs-resourcegroupid>)指定组织和资源集,实现实例资源的隔离查询。

- 组织ID (<x-acs-organizationid>) 不指定则默认获取当前用户所属的组织。
- 资源集ID(<x-acs-resourcegroupid>)不指定默认为空,如果指定资源集需要同时指定组织ID。

6. 查找API的开发文档。

您需要查看目标云产品的开发指南,了解API的功能与使用方式,查找开发指南的方式请参见开发文档 使用说明。

7. 通过SDK调用云产品API。

完成以上准备工作后,您可以使用SDK对API进行调用,调用的方式请根据实际使用的语言,在本手册中查看对应的调用文档。

以通过Java SDK调用ECS的DescribeInstances接口为例,调用的示例代码如下:

```
public static void main(String[] args) {
   ASClient client = new ASClient();
   //指定调用者的标示,仅用于记录日志
   client.setSdkSource("autoTest");
   // 准备请求参数
   Map<String, Object> parameters = new HashMap<String, Object>();
   parameters.put("Product", "dms-enterprise");//与对应API目录中的产品POPCode一致
   parameters.put("Action", "GetOpLog"); //API名称,大小写需要与API目录中的一致
   parameters.put("Version", "2014-05-26"); //API版本
   //填写获取到的RegionID
   parameters.put("RegionId", "xxx");
   //填写获取到AccessKey ID和AccessKey Secret
   parameters.put("AccessKeyId", "xxx");
   parameters.put("AccessKeySecret", "xxx");
   //传入API文档中参数,请注意大小写敏感
   parameters.put("PageNumber",1);
   parameters.put("PageSize",10);
   Map<String, String> headers=new HashMap<>();
   // 输入当前操作的Apsara Uni-manager运营控制台组织ID
   headers.put("x-acs-organizationid","3");
   // 输入当前操作的Apsara Uni-manager运营控制台资源集ID
   //headers.put("x-acs-resourcegroupid", "<your-resourcegroupid>");
   // 填写获取到ASAPI的Endpoint
   String endpoint="https://public.asapi.xxx.xxx/asapi/v3";
   // 发起请求
   String result = client.doPost(endpoint, headers, parameters);
   // 打印结果
   System.out.println(JSONObject.toJSONString(JSONObject.parseObject(result), Serializ
erFeature.PrettyFormat));
```

7.2. 通过ASAPI网关调用专有云API

7.2.1. 获取ASAPI的Endpoint

本文将为您介绍三种获取Endpoint的方法及其校验方法。

获取方法

方法一: 通过服务注册变量获取Endpoint

数据管理 开发指南· <mark>附录</mark>

- 1. 登录飞天基础运维平台。
- 2. 在左侧导航栏中单击报表。
- 3. 在全部报表页面搜索 服务注册变量 , 单击报表名称。



4. 在服务注册变量页面,单击Service旁的 = 图标,搜索 asapi 。



- 5. 在asapi的Service Registration列中,单击鼠标右键,选择显示更多。 在详情页面,查看的asapi.public.endpoint值。
- 6. 按照以下格式生成ASAPI的Endpoint。

https://<asapi.public.endpoint>/asapi/v3

方法二: 通过集群资源生成Endpoint

- 1. 登录飞天基础运维平台。
- 2. 进入集群运维。
 - i. 在左侧导航栏中选择**运维>集群运维**。
 - ii. 在集群文本框中输入 ascm 。
 - iii. 单击操作列中的运维。
- 3. 获取domain值。
 - i. 单击集**群资源**页签。
 - ii. 在名称文本框中输入 asapi_dns_public 。

开发指南· <mark>附录</mark> 数据管理

iii. 单击申请参数栏中的详情。

在申请参数页面查看domain值。

```
{
    "domain": "public.asapi.
    ha_apply_type": "active=active",
    "name": "asapi_dns_public",
    "vip":
    "
}
```

4. 按照以下格式生成ASAPI的Endpoint。

https://<domain>/asapi/v3

方法三: 自行拼装Endpoint

您还可以按照以下格式自行拼装Endpoint。

https://public.asapi.<region>.<internet-domain>/asapi/v3

? 说明

- 1. <region>: 替换为真实的地域ID值。
- 2. <internet-domain>:替换为专有云的根域名(特别提示是internet,不是intranet),可咨询专有云运维管理员。

假设region示例值为 cn-****-***-d01 , internet-domain示例值为 inter.***.***.com 。

完整的Endpoint示例值如下:

```
https://public.asapi.cn-***-***-d01.inter.***.***.com/asapi/v3
```

Endpoint (业务域名) 校验

输入以下代码, 若访问异常则表示业务域名不可用, 可使用历史业务域名。

curl https://public.asapi.\${global:region}.\${global:internet-domain}/api/version -k

? 说明

\${global:region} 和 \${global:internet-domain} 为变量,需要替换为您获取到的实际值。

数据管理 开发指南·附录

访问正常示例:

```
root@ - - # curl https://public.asapi. . _______ :/api/version -k
{"version":"v3.13","code":"200","apiCount":12379,"desc":"update xml pretty"}root@ root@
```

访问异常示例:

```
[root@color=land | policy | po
```

历史业务域名

仅支持HTTP协议,业务域名后缀为 intranet-domain ,业务域名结构如下:

http://server.asapi.\${global:region}.\${global:intranet-domain}/asapi/v3



操作步骤

获取变量\${global:region}、\${global:internet-domain}、\${global:intranet-domain}实际值的操作步骤如下:

1. 在地址栏中,输入Apsara Uni-manager运维控制台的访问地址*region-id*.aso.*intranet-domain-id*.com,按回车键。



? 说明

您可以单击页面右上角的下拉按钮来进行语言切换。

2. 输入正确的用户名及密码。

? 说明

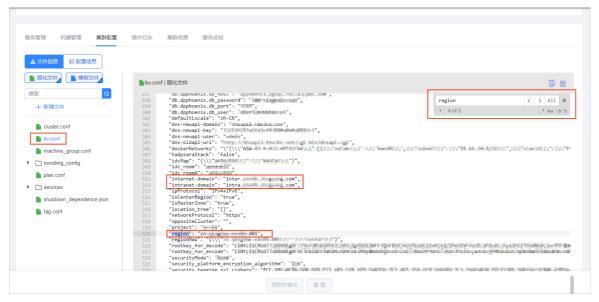
登录Apsara Uni-manager运维控制台的用户名和密码请从部署人员或管理员处获取。

首次登录Apsara Uni-manager运维控制台时,需要修改登录用户名的密码,请按照提示完成密码修改。为提高安全性,密码必须满足以下要求:

- 。 英文大小写字母
- 阿拉伯数字 (0~9)
- 特殊符号,包括感叹号(!)、at(@)、井号(#)、美元符号(\$)、百分号(%)等
- 密码长度10~20个字符
- 3. 单击登录,进入Apsara Uni-manager运维控制台页面。
- 4. 在左侧菜单栏中,选择运维>集群运维。
- 5. 在集群运维页面搜索ascm-A,单击集群名称。



- 6. 单击集群配置页签。
- 7. 在**集群配置**页签左侧**文件列**表中,单击**kv.conf**,单击右侧文件内容,按Ctrl+F键分别搜索region、internet-domain和intranet-domain。



获取到的region值将替换 \${global:region} , internet-domain值将替换

\${global:internet-domain} , intranet-domain值将替换 \${global:intranet-domain} 。

 数据管理 开发指南· <mark>附录</mark>

? 说明

最终Endpoint (业务域名)中, \${global:} 不保留。

7.2.2. 获取ASAPI的Endpoint

本文将为您介绍获取Endpoint的方法。

- 1. 登录OPS1机器。
- 2. 执行以下命令,获取三个不同的域名。

```
kubectl get ingress -n ascm |grep asapiserver
```

3. 选取 public 开头的域名,作为 domain 值。



4. 按照以下格式生成ASAPI的Endpoint。

https://<domain>/asapi/v3

7.2.3. 获取ASAPI的SDK包

专有云环境中使用ASAPI作为所有接入ASAPI产品的统一入口,并提供SDK进行产品侧接口的调用支持。

公网环境

公网环境下您可以进入专有云文档官网,在页面底部的其他资源中,下载对应版本ASAPI的SDK包。

非公网环境

若在非公网环境下使用ASAPI的SDK包,需要先在公网环境下进入<mark>专有云文档官网</mark>,在页面底部的**其他资源**中,下载对应版本ASAPI的SDK包和ASAPI的SDK依赖包。

ASAPI的SDK依赖包(JAVA)

greopld	artifactId	version
com.alibaba	alicrypto-java-aliyun	1.0.4
com.alibaba	fastjson	1.2.68
org.bouncycastle	bcprov-jdk15to18	1.66
org.bouncycastle	bcpkix-jdk15on	1.65
org.bouncycastle	bcprov-jdk15on	1.65
com.aliyun	gmsse	1.1.0
org.apache.commons	commons-lang3	3.0

开发指南· <mark>附录</mark> 数据管理

greopld	artifactId	version
commons-codec	commons-codec	1.11
commons-logging	commons-logging	1.2
org.apache.httpcomponents	httpcore	4.4.11
org.apache.httpcomponents	httpclient	4.5.7

ASAPI的SDK依赖包 (Python)

Package	Version
aliyun-python-sdk-core	2.13.35
certifi	2021.5.30
cffi	1.14.6
charset-normalizer	2.0.4
cryptography	3.4.8
idna	3.2
jmespath	0.10.0
pycparser	2.20
requests	2.26.0
urllib3	1.26.6

? 说明 Package版本支持使用更高或最新版本。

7.3. STS概要

7.3.1. 什么是STS

阿里云STS(Security Token Service)是阿里云提供的一种临时访问权限管理服务。RAM提供RAM用户和RAM角色两种身份。其中,RAM角色不具备永久身份凭证,而只能通过STS获取可以自定义时效和访问权限的临时身份凭证,即安全令牌(STS Token)。

使用RAM用户扮演角色时获取STS Token

有权限的RAM用户可以使用其访问密钥调用AssumeRole接口,以获取某个RAM角色的STS Token,从而使用STS Token访问阿里云资源。

产品优势

数据管理 开发指南· <mark>附录</mark>

- 使用STS Token,减少长期访问密钥(Accesskey)泄露的风险。
- STS Token具有时效性,可以自定义有效期,到期后将自动失效,无需定期轮换。

● 可以为STS Token绑定自定义权限策略,提供更加灵活和精细的云资源授权。

7.3.2. AssumeRole

RAM用户调用AssumeRole接口获取一个扮演RAM角色的临时身份凭证(STS Token)。该接口中RAM用户扮演的是可信实体为阿里云账号类型的RAM角色。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	AssumeRole	要执行的操作。取值:AssumeRole
RoleArn	String	是	acs:ram::1234567 89012***:role/ad minrole	要扮演的RAM角色ARN。 该角色是可信实体为阿里云账号类型的RAM 角色。
RoleSessionNam e	String	是	alice	角色会话名称。 该参数为用户自定义参数。通常设置为调用该API的用户身份,例如:用户名。在操作审计日志中,即使是同一个RAM角色执行的操作,也可以根据不同的RoleSessionName来区分实际操作者,以实现用户级别的访问审计。 长度为2~64个字符,可包含英文字母、数字、半角句号(.)、at(@)、短划线(-)和下划线(_)。
Policy	String	否	{"Statement": [{"Action": ["*"],"Effect": "Allow","Resource ": ["*"]}],"Version":"1	为STS Token额外添加的一个权限策略,进一步限制STS Token的权限。具体如下:长度为1~1024个字符。

返回结果

名称		类型	示例值	描述
----	--	----	-----	----

名称	类型	示例值	描述
RequestId	String	6894B13B-6D71- 4EF5-88FA- F32781734A7F	请求ID。
Credentials	无	无	访问凭证。
AccessKeyId	String	STS.L4aBSCSJVMuKg5 U1****	访问密钥ID。
AccessKeySecret	String	wyLTSmsyPGP1ohvv w8xYgB29dlGl8KMiH2 pK****	访问密钥。
SecurityT oken	String	*****	安全令牌。
Expiration	String	2015-04- 09T11:52:19Z	失效时间。
AssumedRoleUser	无	无	角色扮演时的临时身份。
Arn	String	acs:ram::123456789 012****:role/adminr ole/alice	临时身份的ARN。
AssumedRoleId	String	34458433936495**** : alice	临时身份的ID。

示例

引入Pom依赖

操作需要依赖阿里云核心库来实现,在Pom中引入阿里云核心库包的操作如下:

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>[4.4.9,5.0.0)</version>
</dependency>
```

调用API获取STS AccessKey三元组

数据管理 开发指南· <mark>附录</mark>

以下代码示例为通过调用 Sts::AssumeRole::2015-04-01 获取STS AccessKey三元组信息。

```
package com.aliyuns.test;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpClientConfig;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;
import org.apache.commons.lang3.StringUtils;
public class STSAccessKeyDemo {
   public static void main(String[] args) {
       // 创建DefaultAcsClient实例并初始化。
       DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourAccessKe
yID>", "<yourAccessKeySecret>");
       HttpClientConfig clientConfig = HttpClientConfig.getDefault();
       clientConfig.setIgnoreSSLCerts(true);
       clientConfig.setProtocolType(ProtocolType.HTTPS);
       profile.setHttpClientConfig(clientConfig);
       IAcsClient client = new DefaultAcsClient(profile);
       CommonRequest request = new CommonRequest();
       // "sts.aliyuns.com"要替换为实际的STS Endpoint, 一般格式为sts-vpc.${global:region}.${g
lobal:internet-domain}
       request.setSysDomain("sts.aliyuns.com");
       request.setSysProduct("Sts");
       request.setSysAction("AssumeRole");
        request.setSysVersion("2015-04-01");
       request.setSysMethod(MethodType.POST);
       request.setHttpContentType(FormatType.FORM);
       // 业务参数,设置本次调用Session的别名,"<yourRoleSessionName>"请替换为实际值
        request.putQueryParameter("RoleSessionName", "<yourRoleSessionName>");
        // 扮演的Role角色ARN,"<yourRoleArn>"请替换为上一步获取到的实际Ram Role ARN
       request.putQueryParameter("RoleArn", "<yourRoleArn>");
       try {
            CommonResponse response = client.getCommonResponse(request);
           String assumeRole = response.getData();
           if (StringUtils.isEmpty(assumeRole)) return;
           JSONObject assumeRoleJson = JSONObject.parseObject(assumeRole);
           JSONObject credentials = assumeRoleJson.getJSONObject("Credentials");
            // 获取到STS三元组
           String stsAK = credentials.getString("AccessKeyId");
           String stsSK = credentials.getString("AccessKeySecret");
            String stsToken = credentials.getString("SecurityToken");
           System.out.println("stsAk: " + stsAK +"\nstsSk: " + stsSK + "\nstsToken: " + st
sToken);
        } catch (ServerException e) {
```

开发指南·<mark>附录</mark> 数据管理

```
System.out.println(e.getErrCode());
System.out.println(e.getErrMsg());
} catch (ClientException e) {
    System.out.println(e.getErrCode());
    System.out.println(e.getErrMsg());
}
}
```

调用示例

下列代码以调用ECS的 DescribeZones 接口为示例。

数据管理 开发指南· <mark>附录</mark>

```
package com.aliyuns.test;
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpClientConfig;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;
public class DescribeZonesDemo {
   public static void main(String[] args) {
       // "<yourRegionID>"替换为真实的RegionID
       // "<yourStsAccesskey>", "<yourStsAccessKeySecret>"替换为上面获取到的STS三元组中stsAk、
stsSk
       DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourStsAcces
sKey>", "<yourStsAccessKeySecret>");
       HttpClientConfig clientConfig = HttpClientConfig.getDefault();
       clientConfig.setIgnoreSSLCerts(true);
       clientConfig.setProtocolType(ProtocolType.HTTPS);
       profile.setHttpClientConfig(clientConfig);
       IAcsClient client = new DefaultAcsClient(profile);
       CommonRequest request = new CommonRequest();
       request.setSysDomain("ecs.aliyuns.com");
        request.setSysProduct("Ecs");
       request.setSysAction("DescribeZones");
       request.setSysVersion("2014-05-26");
       request.setSysMethod(MethodType.POST);
        request.setHttpContentType(FormatType.FORM);
       //"<yourRegionID>"替换为真实的RegionId
       request.putHeadParameter("x-acs-regionid", "<yourRegionID>");
       // "<yourStsToken>"替换为上面获取到的STS三元组中stsToken
        request.putQueryParameter("SecurityToken", "<yourStsToken>");
       // 非必填,资源集id
       // request.putHeadParameter("x-acs-resourcegroupid", "<yourResourceGroupID>");
       // 必填,组织id
        request.putHeadParameter("x-acs-organizationid", "<yourOrganizationID>");
       trv {
           CommonResponse response = client.getCommonResponse(request);
           System.out.println(response.getData());
        } catch (ServerException e) {
           System.out.println(e.getErrCode());
           System.out.println(e.getErrMsg());
        } catch (ClientException e) {
           System.out.println(e.getErrCode());
           System.out.println(e.getErrMsg());
```

开发指南·<mark>附录</mark> 数据管理

错误码

HttpCode	错误码	错误信息	描述
400	InvalidParameter	The parameter RoleArn is wrongly formed.	角色ARN格式错误。
400	InvalidParameter.RoleAr n	The parameter RoleArn is wrongly formed.	角色ARN格式错误
400	InvalidParameter.RoleSe ssionName	The parameter RoleSessionName is wrongly formed.	RoleSessionName格式错误,支持输入2~64个字符,请输入至少2个字符,允许输入英文字母、数字、半角句号(.)、at(@)、短划线(-)和下划线(_)。
400	InvalidParameter.Durati onSeconds	The Min/Max value of DurationSeconds is 15min/1hr.	DurationSeconds参数设置错误,过期时间最小值为900秒,最大值为MaxSessionDuration设置的时间。
400	InvalidParameter.PolicyG rammar	The parameter Policy has not passed grammar check.	权限策略语法错误。
400	InvalidParameter.PolicyS ize	The size of Policy must be smaller than 1024 bytes.	权限策略长度超限,最大 不超过1024字符。
403	NoPermission	You are not authorized to do this action. You should be authorized by RAM.	STS Token没有权限。
404	EntityNotExist.Role	The specified Role not exists.	指定的RAM角色不存在。
500	InternalError	STS Server Internal Error happened.	服务器内部错误。