

ALIBABA CLOUD

阿里云

专有云企业版

云数据库 Redis 版
开发指南

产品版本：v3.16.2

文档版本：20220915



法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1. ASAPI 开发指南	07
1.1. 准备工作	07
1.1.1. 登录 API 与工具控制台	07
1.1.2. 查看 API 信息	07
1.1.3. 获取 AccessKey	08
1.1.4. 获取公共 Header 参数	10
1.1.5. 获取 STS AccessKey	11
1.2. 通过 POP 网关调用专有云 API	13
1.2.1. 调用前置	13
1.2.2. Java 版	14
1.2.3. Go 版	17
1.2.4. Python 版	20
1.3. 通过 ASAPI 网关调用专有云 API	22
1.3.1. ASAPI 概述	22
1.3.2. 生成 ASAPI 的 Endpoint	23
1.3.3. 获取 ASAPI 的 SDK 包	25
1.3.4. 公共参数	25
1.3.5. SDK 说明	26
1.3.5.1. Java SDK 调用	26
1.3.5.2. Python SDK 调用	31
1.3.5.3. Node SDK 调用	33
1.3.6. HTTP 调用方式	35
1.3.6.1. 概述	35
1.3.6.2. 请求结构	35
1.3.6.3. 签名机制	36
1.3.6.4. 返回结果	40

2.API 参考	42
2.1. API 概览	42
2.2. 请求结构	43
2.3. 公共参数	43
2.4. 返回结果	44
2.5. 签名机制	46
2.6. 如何使用RAM授权	47
2.6.1. 使用 RAM 实现资源授权	47
2.6.2. RAM 中可授权的资源类型	48
2.6.3. RAM 中可对资源进行授权的 Action	48
2.6.4. API 的鉴权规则	49
2.7. 生命周期管理	50
2.7.1. CreateInstance	50
2.7.2. DeleteInstance	52
2.7.3. ModifyInstanceSpec	53
2.8. 实例管理	55
2.8.1. DescribeInstanceAttribute	55
2.8.2. DescribeInstances	59
2.8.3. DescribeRegions	64
2.8.4. DescribeSecurityIps	66
2.8.5. FlushInstance	68
2.8.6. ModifyInstanceAttribute	69
2.8.7. ModifyInstanceMaintainTime	70
2.8.8. ModifyInstanceNetExpireTime	71
2.8.9. ModifySecurityIps	73
2.8.10. SwitchNetwork	74
2.9. 备份恢复	75
2.9.1. CreateBackup	75

2.9.2. ModifyBackupPolicy	76
2.9.3. DescribeBackupPolicy	78
2.9.4. DescribeBackups	79
2.9.5. RestoreInstance	83
2.10. 监控管理	84
2.10.1. DescribeMonitorItems	84
2.10.2. DescribeHistoryMonitorValues	86
2.11. 参数管理	88
2.11.1. DescribeInstanceConfig	88
2.11.2. ModifyInstanceConfig	90
2.12. 附表	91
2.12.1. 实例规格表	91
2.12.2. 错误码表	96
2.12.3. 实例配置参数表	100

1.ASAPI开发指南

1.1. 准备工作

1.1.1. 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。

 **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**账号登录**。
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**账号登录**。
 - 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**。

 **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**。

1.1.2. 查看API信息

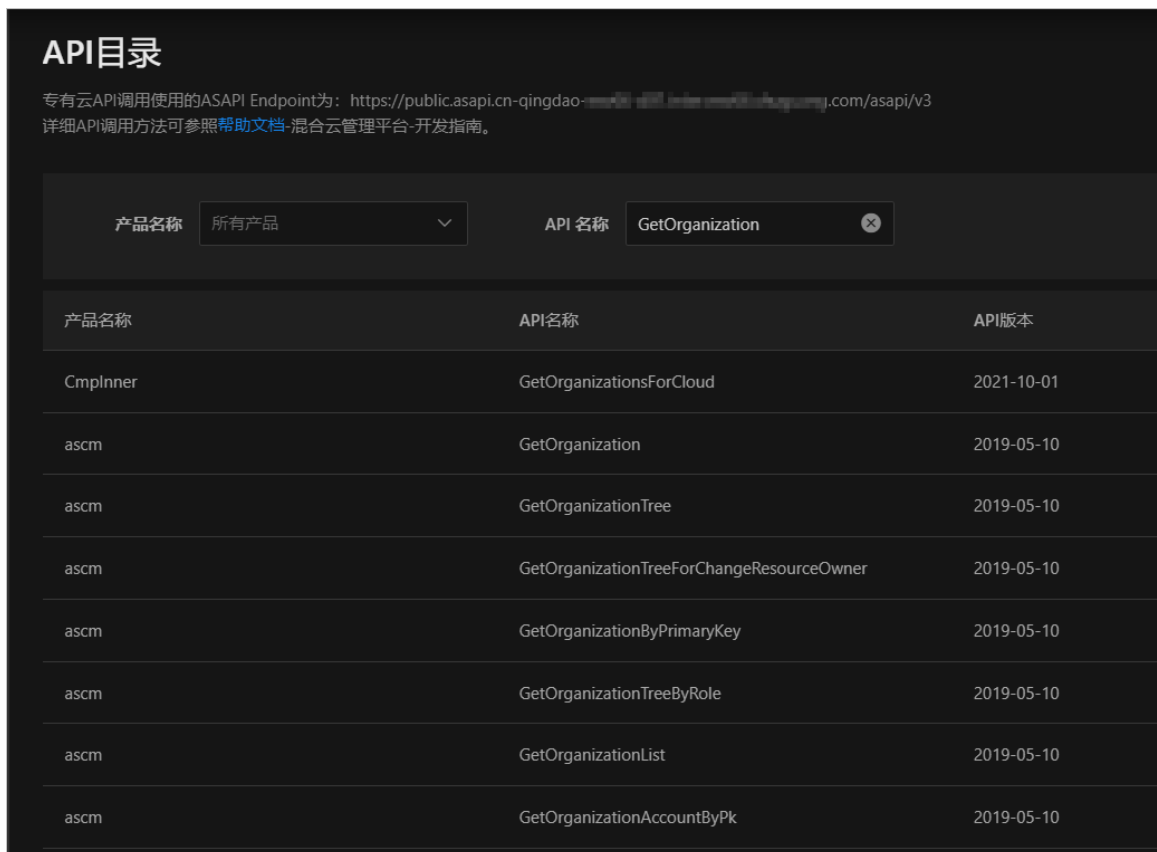
您需要获取产品名称、API名称和API版本，使用SDK时需要填写这些信息。

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。

1.1.3. 获取AccessKey

访问身份验证中通过使用AccessKey ID和AccessKey Secret对称加密的方法来验证某个请求的发送者身份。AccessKey ID用于标示用户，AccessKey Secret是用户用于加密签名的字符串。本文将为您描述如何获取通过RAM授权的AccessKey。

前提条件

只有运营管理员和一级组织管理员可以获取组织AccessKey。

背景信息

AccessKey支持两种授权模式：


- 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。

本文仅为您介绍如何获取通过RAM授权的AccessKey，通过STS授权的AccessKey获取方式请参见[获取STS 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：


参数名称	描述
x-acs-regionid	环境Region，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Apsara Uni-manager运营控制台开发指南中的GetOrganizationList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Apsara Uni-manager运营控制台开发指南中的ListResourceGroup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。


 **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。

获取个人账号AccessKey

获取个人账号AccessKey的方法如下：

- 登录Apsara Uni-manager运营控制台。
- 在系统界面右上角，将鼠标悬停在当前登录用户的头像上，单击**个人信息**。
- 在**阿里云AccessKey**区域，您可以查看个人账户的AccessKey信息。

AccessKey + 创建AccessKey				
 AccessKey ID和AccessKey Secret是您访问云服务资源时的密钥，具有该账号完整的权限，请您妥善保管。 ×				
AccessKey ID	AccessKey Secret	状态	创建时间	
z*****w	*****	 启用	2022年01月13日 20:45:35	

 **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：

- 管理员登录Apsara Uni-manager运营控制台。
- 在顶部菜单栏，单击**企业**。
- 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。

1.1.4. 获取公共Header参数

调用API时需要提供多个Header参数。本文为您介绍如何获取这些参数，并对调用后的一些返回值做出解释。

获取Region ID

1. 管理员登录Apsara Uni-manager运营控制台。
2. 在顶部菜单栏中，单击**企业**。
3. 在左侧导航栏中，选择**资源管理>地域管理**。
4. 单击目标组织名称。
5. 在右侧**Region**列表中查看Region ID。

获取组织ID

组织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[Get OrganizationList](#)。

获取资源集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*中的章节[List ResourceGroup](#)。

获取实例ID

请各云产品自行修改替换产品名及获取实例ID流程

实例ID可以通过Apsara Uni-manager运营控制台中的云产品实例列表进行获取，本文中以ECS为例，获取步骤如下：

1. 登录Apsara Uni-manager运营控制台。
2. 在顶部的菜单栏中，选择**产品>弹性计算>云服务器ECS**。
3. 在左侧导航栏中，单击**实例**。
4. 在实例列表中查看**实例ID**。

返回参数

对于列表类的云产品资源实例接口，调用后会默认增加如下返回值。

名称	描述
ResourceGroupName	资源集名称。
ResourceGroup	资源集ID。

名称	描述
DepartmentName	组织名称。
Department	组织ID。

1.1.5. 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。您可以选择利用STS方式，调用专有云的产品API。

操作步骤

1. 获取用户角色的 `RAM Role` 。

- 登录Apsara Uni-manager运营控制台。
- 单击页面右上角的用户头像，选择个人信息。
- 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. 使用以下代码，通过角色的 `RAM Role` 获得 `stsAK` 、 `stsSK` 和 `stsToken` 。

```
public static void main(String[] args) {
    ASClient asClient = new ASClient();
    // 调用来源说明
    asClient.setSdkSource("asapi_test");
    // 传入Arn
    String assumeRole = AssumeRole(asClient, "<RAM Role>");

    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");
}

public static String AssumeRole(ASClient asClient, String roleArn) {

    Map<String, String> headers = new HashMap<>();

    Map<String, Object> reqParams = new HashMap<>();

    reqParams.put("action", "AssumeRole");

    reqParams.put("Product", "Sts");

    reqParams.put("Version", "2015-04-01");

    reqParams.put("regionId", regionId);

    // 当前登录用户的ak
    reqParams.put("AccessKeyId", <当前用户的AccessKey ID>);
    // 当前登录用户的sk
    reqParams.put("AccessKeySecret", <当前用户的AccessKey Secret>);

    reqParams.put("roleSessionName", "AsapiStsUser");

    reqParams.put("roleArn", roleArn);

    return asClient.doRequest(endpoint, headers, reqParams);
}
```

3. 通过STS可以调用云产品的接口，下列代码以ECS为示例。

```
public static void main(String[] args) {
    ASClient asClient = new ASClient();
    asClient.setSdkSource("asapi_test");
    // 调用云产品的API
    String ecsRes = DescribeZones(asClient,<stsAK>,<stsSK>,<stsToken>);
}

public static String DescribeZones(ASClient asClient, String stsAK, String stsSK, String stsToken) {

    Map<String, String> headers = new HashMap<>();

    Map<String, Object> reqParams = new HashMap<>();

    reqParams.put("action", "DescribeZones");

    reqParams.put("Product", "Ecs");

    reqParams.put("Version", "2014-05-26");

    reqParams.put("regionId", "<your-RegionID>");

    //STS AK/SK/Token三元组
    reqParams.put("AccessKeyId", <stsAK>);
    reqParams.put("AccessKeySecret", <stsSK>);
    reqParams.put("SecurityToken", <stsToken>);

    return asClient.doRequest(endpoint, headers, reqParams, HttpMethod.GET);
}
```

1.2. 通过POP网关调用专有云API

1.2.1. 调用前置

阿里云专有云与公有云一样，可通过POP网关的方式进行API调用。本文为您介绍调用前需要进行的准备工作。

背景信息

阿里云专有云既支持通过ASAPI网关调用专有云API，也支持通过POP网关调用专有云API。为了使专有云保持与公有云相同的使用方式，使您拥有更方便，更一致的体验，建议通过POP网关调用专有云API。

获取云产品的Endpoint

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。

您也可以自行获取Endpoint，本文以ECS为例为您介绍获取云产品业务域Endpoint的方法。

1. 登录飞天基础运维平台。
 - i. 登录Apsara Uni-manager运维控制台。

- ii. 顶部菜单栏选择产品运维 > 云平台运维 > 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 ECS。
5. 在 ecs-yaochi 服务的Service Registration列中，单击鼠标右键，选择显示更多。

在详情页面，查看的internal.openapi.endpoint值，即 Endpoint。

获取公共参数

通过POP网关调用专有云API过程中所使用的参数，如组织ID、资源集ID、RegionID和实例ID，获取方式请参见[获取公共Header参数](#)。

1.2.2. Java版

通过POP Java SDK访问产品域名，需要了解该产品调用方式(RPC或ROA)。

前提条件

- 通过[阿里云开发工具包 \(SDK\)](#) 下载对应产品的安装依赖。
- HTTPS访问设置为忽略SSL证书验证。

调用示例

RPC调用，示例接口信息：产品（ECS）、接口（DescribeInstances）、版本（2014-05-26）。

调用示例如下：

```
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.MethodType;
import com.aliyuncs.profile.DefaultProfile;

public class DescribeInstances {

    public static void main(String[] args) {
        // 创建DefaultProfile实例并初始化
```

```
// 创建DefaultAcsClient实例并初始化。
DefaultProfile profile = DefaultProfile.getProfile("<your-regionid>", "<yourAccessKeyId>", "<yourAccessSecret>");

// 忽略SSL
// HttpClientConfig clientConfig = HttpClientConfig.getDefault();
// clientConfig.setIgnoreSSLCerts(true);
// 设置连接超时
// clientConfig.setConnectionTimeoutMillis(3000L);
// 设置读取超时
// clientConfig.setReadTimeoutMillis(10000L);
// profile.setHttpClientConfig(clientConfig);

IAcsClient client = new DefaultAcsClient(profile);
// 创建API请求并设置参数。
CommonRequest request = new CommonRequest();
// 设置请求方式
request.setSysMethod(MethodType.POST);
// 设置云产品Endpoint
request.setSysDomain("<ecsEndpoint>");
// 设置访问接口版本
request.setSysVersion("2014-05-26");
// 设置LocationProduct
request.setSysLocationProduct("Ecs");
// 设置访问接口的名称
request.setSysAction("DescribeInstances");

// 当使用sts ak调用RPC接口时 需要在参数中加上SecurityToken字段
// request.putBodyParameter("SecurityToken", "<your-SecurityToken>");

// 业务参数
request.putBodyParameter("PageSize", "10");
request.putBodyParameter("PageNumber", "1");

// 必填值，调用来源说明
request.putHeadParameter("x-acs-caller-sdk-source", "ApiTestDemo");
// 非必填，企业版V3.15.0之前历史版本需要
request.putHeadParameter("x-acs-regionid", "<your-regionid>");
// 非必填，资源集ID
request.putHeadParameter("x-acs-resourcegroupid", "<your-resourcegroupid>");
// 必填 组织ID
request.putHeadParameter("x-acs-organizationid", "<your-organizationid>");
// 非必填，修改&删除操作需要
// request.putHeadParameter("x-acs-instanceid", "<your-instanceid>");

try {
    // 发起请求并处理应答或异常。
    CommonResponse response = client.getCommonResponse(request);
    System.out.println(response.getData());
} 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());
}
```

```

        System.out.println("requestId: " + e.getRequestId());
    }

}

}

```

ROA调用，示例接口信息：产品（CR）接口（GetNamespaceList）版本（2016-06-07）。

调用示例如下：

```

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.MethodType;
import com.aliyuncs.profile.DefaultProfile;

public class GetNamespaceList {

    public static void main(String[] args) {
        // 创建DefaultAcsClient实例并初始化。
        DefaultProfile profile = DefaultProfile.getProfile("<your-regionid>", "<yourAccessKeyId>", "<yourAccessSecret>");

        // 忽略SSL
        // HttpClientConfig clientConfig = HttpClientConfig.getDefault();
        // clientConfig.setIgnoreSSLCerts(true);
        // 设置连接超时
        // clientConfig.setConnectionTimeoutMillis(3000L);
        // 设置读取超时
        // clientConfig.setReadTimeoutMillis(10000L);
        // profile.setHttpClientConfig(clientConfig);

        IAcsClient client = new DefaultAcsClient(profile);
        // 创建API请求并设置参数。
        CommonRequest request = new CommonRequest();

        // 设置HTTPS访问
        request.setSysProtocol(ProtocolType.HTTP);
        // 设置请求方式
        request.setSysMethod(MethodType.GET);
        // 设置云产品endpoint
        request.setSysDomain("<crEndpoint>");
        // 设置访问接口版本
        request.setSysVersion("2016-06-07");
        // 设置uriPattern 只有roa调用需要传递此字段
        request.setSysUriPattern("/namespace");
        // 设置LocationProduct
        request.setSysLocationProduct("cr");
        // 设置 访问接口的名称
        request.setSysAction("GetNamespaceList");

        // 必填值，调用来源说明
    }
}

```



```
request.putHeadParameter("x-acs-caller-sdk-source", "ApiTestDemo");
// 非必填，企业版V3.15.0之前历史版本需要
request.putHeadParameter("x-acs-regionid", "<your-regionid>");
// 非必填，资源集ID
request.putHeadParameter("x-acs-resourcegroupid", "<your-resourcegroupid>");
// 必填 组织ID
request.putHeadParameter("x-acs-organizationid", "<your-organizationid>");
// 非必填，修改&删除操作需要
// request.putHeadParameter("x-acs-instanceid", "<your-instanceid>");

// 当使用STS AK调用ROA接口时 需要在Header加上x-acs-security-token字段
// request.putHeadParameter("x-acs-security-token", "<your-securitytoken>");

try {
    // 发起请求并处理应答或异常。
    CommonResponse response = client.getCommonResponse(request);
    System.out.println(response.getData());
} 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());
}
}
```

注意

设置HTTPS访问，一定要设置忽略SSL证书验证。

1.2.3. Go版

本文为您介绍POP Go SDK调用的方式。

前提条件

通过[阿里云开发工具包（SDK）](#)下载对应产品的安装依赖。

背景信息

当前支持RPC和ROA的方式调用API。

调用示例

RPC方式调用示例接口信息：产品（ECS）、接口（DescribeInstances）、版本（2014-05-26）。

示例代码如下：

```
package main

import (
    "fmt"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
)

func main() {
    client, err := sdk.NewClientWithAccessKey("<your-RegionID>", "<AccessKey ID>", "<Access Key Secret>")

    request := requests.NewCommonRequest()

    request.Method = "POST"
    request.Domain = "<ECSEndpoint>"
    request.Version = "2014-05-26"
    request.ApiName = "DescribeInstances"
    request.Product = "Ecs"
    // 当使用sts ak调用rpc接口时 需要在参数中加上SecurityToken字段
    // request.QueryParams["SecurityToken"]="<your-SecurityToken>"

    // 必填值, 调用来源说明
    request.Headers["x-acs-caller-sdk-source"] = "ApiTestDemo"
    // 非必填, 企业版v3.15.0之前历史版本需要
    request.Headers["x-acs-regionid"] = "<your-RegionID>"
    // 非必填, 资源集ID
    request.Headers["x-acs-resourcegroupid"] = "<your-ResourceGroupID>"
    // 必填, 组织ID
    request.Headers["x-acs-organizationid"] = "<your-OrganizationID>"
    // 非必填, 修改&删除操作需要
    // request.Headers["x-acs-instanceid"] = "<your-InstanceID>"

    response, err := client.ProcessCommonRequest(request)
    if err != nil {
        panic(err)
    }
    fmt.Print(response.GetHttpContentString())
}
```

ROA调用示例接口信息：产品（CR）接口（GetNamespaceList）版本（2016-06-07）。

调用示例如下：

```
package main

import (
    "fmt"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
    "os"
)

func main() {
    client, err := sdk.NewClientWithAccessKey("<your-regionid>", "<yourAccessKeyId>", "<yourAccessSecret>")
    request := requests.NewCommonRequest()

    request.Method = "GET"
    request.Domain = "<crEndpoint>"
    request.Version = "2016-06-07"
    request.ApiName = "GetNamespaceList"
    request.Product = "cr"

    request.PathPattern = "/namespace"
    // 当使用sts ak调用ROA接口时 需要在header中加上x-acis-security-token字段
    // request.Headers["x-acis-security-token"] = "<your-SecurityToken>"
    // 必填值, 调用来源说明
    request.Headers["x-acis-caller-sdk-source"] = "ApiTestDemo"
    // 非必填, 企业版v3.15.0之前历史版本需要
    request.Headers["x-acis-regionid"] = "<your-RegionID>"
    // 非必填, 资源集ID
    request.Headers["x-acis-resourcegroupid"] = "<your-ResourceGroupID>"
    // 必填, 组织ID
    request.Headers["x-acis-organizationid"] = "<your-OrganizationID>"
    // 非必填, 修改&删除操作需要
    // request.Headers["x-acis-instanceid"] = "<your-InstanceID>"

    response, err := client.ProcessCommonRequest(request)
    if err != nil {
        panic(err)
    }
    fmt.Print(response.GetHttpContentString())
}
```

设置代理

当您在调用实例时, 或许会需要设置代理, 您可以通过以下两种方式设置代理。

- 通过环境变量设置。

环境变量HTTP_PROXY、HTTPS_PROXY或者NO_PROXY。

设置代码如下:

```
import (
    "os"
)

func main(){
    os.Setenv("HTTP_PROXY", "http://x.x.x.x:xxxx") // 设置 HTTP_PROXY 环境变量
    os.Setenv("HTTPS_PROXY", "https://x.x.x.x:xxxx") // 设置 HTTPS_PROXY 环境变量
}
```

- 通过客户端设置。

设置代码如下：

```
// 客户端设置代理优先级比环境变量高
client.SetHttpProxy("http://x.x.x.x:xxxx") // 设置http代理
client.GetHttpProxy()                       // 获取http代理

client.SetHttpsProxy("https://x.x.x.x:xxxx") // 设置https代理
client.GetHttpsProxy()                       // 获取https代理

client.SetNoProxy("127.0.0.1,localhost")    // 设置代理白名单
client.GetNoProxy()                         // 获取代理白名单
```

1.2.4. Python版

本文为您介绍POP Python SDK调用云产品API的方式。

前提条件

通过[阿里云开发工具包（SDK）](#)下载对应产品的安装依赖。

背景信息

目前仅支持通过RPC方式调用API。

调用示例

调用示例接口信息：产品（ECS）、接口（DescribeInstances）、版本（2014-05-26）。

示例代码如下：

```
from aliyunsdkcore.client import AcsClient
import os
from aliyunsdkcore.request import CommonRequest

# 创建AcsClient实例
client = AcsClient(
    "<AccessKey ID>",
    "<AccessKey Secret>",
    "<your-RegionID>",
    timeout=1000, # 设置超时时间
    verify=False, # 忽略证书 或者 verify='<证书位置>'
);
# 和https一起使用
# client.set_verify(False)
# os.environ["HTTPS_PROXY"] = "http://127.0.0.1:8080"
```

```
# os.environ["HTTP_PROXY"] = "http://127.0.0.1:8080"
def hello_ecs_instance():
    # 创建request，并设置参数
    request = CommonRequest()

    # 当使用sts ak调用rpc接口时 需要在参数中加上SecurityToken字段
    # params = {"PageSize": "10", "PageNumber": "1", "SecurityToken": "<your-SecurityToken>"};
    params = {"PageSize": "10", "PageNumber": "1"};
    request.set_body_params(params)

    headers = {
        # 必填值，调用来源说明
        "x-acs-caller-sdk-source": "ApiTestDemo",
        # 非必填，企业版V3.15.0之前历史版本需要
        "x-acs-regionid": "<your-RegionID>",
        # 非必填，资源集ID
        "x-acs-resourcegroupid": "<your-ResourceGroupID>",
        # 必填，组织ID
        "x-acs-organizationid": "<your-OrganizationID>"
        # 非必填，修改&删除操作需要
        # "x-acs-instanceid": "<your-InstanceID>"
    }
    request.set_headers(headers)
    request.set_version("2014-05-26")
    request.set_action_name("DescribeInstances")
    request.set_endpoint("<ecsEndpoint>") //Endpoint请替换为真实地址
    request.set_product("Ecs")
    request.set_method("POST")

    # 发起API请求并显示返回值
    response = client.do_action_with_exception(request)
    print(response)

if __name__ == '__main__':
    print("Hello Aliyun OpenApi!")
    hello_ecs_instance()
```

设置代理

调用API时您或许需要通过环境变量设置HTTP_PROXY、HTTPS_PROXY。

示例代码如下：

```
# https的代理
os.environ["HTTPS_PROXY"] = "http://x.x.x.x:xxxx"

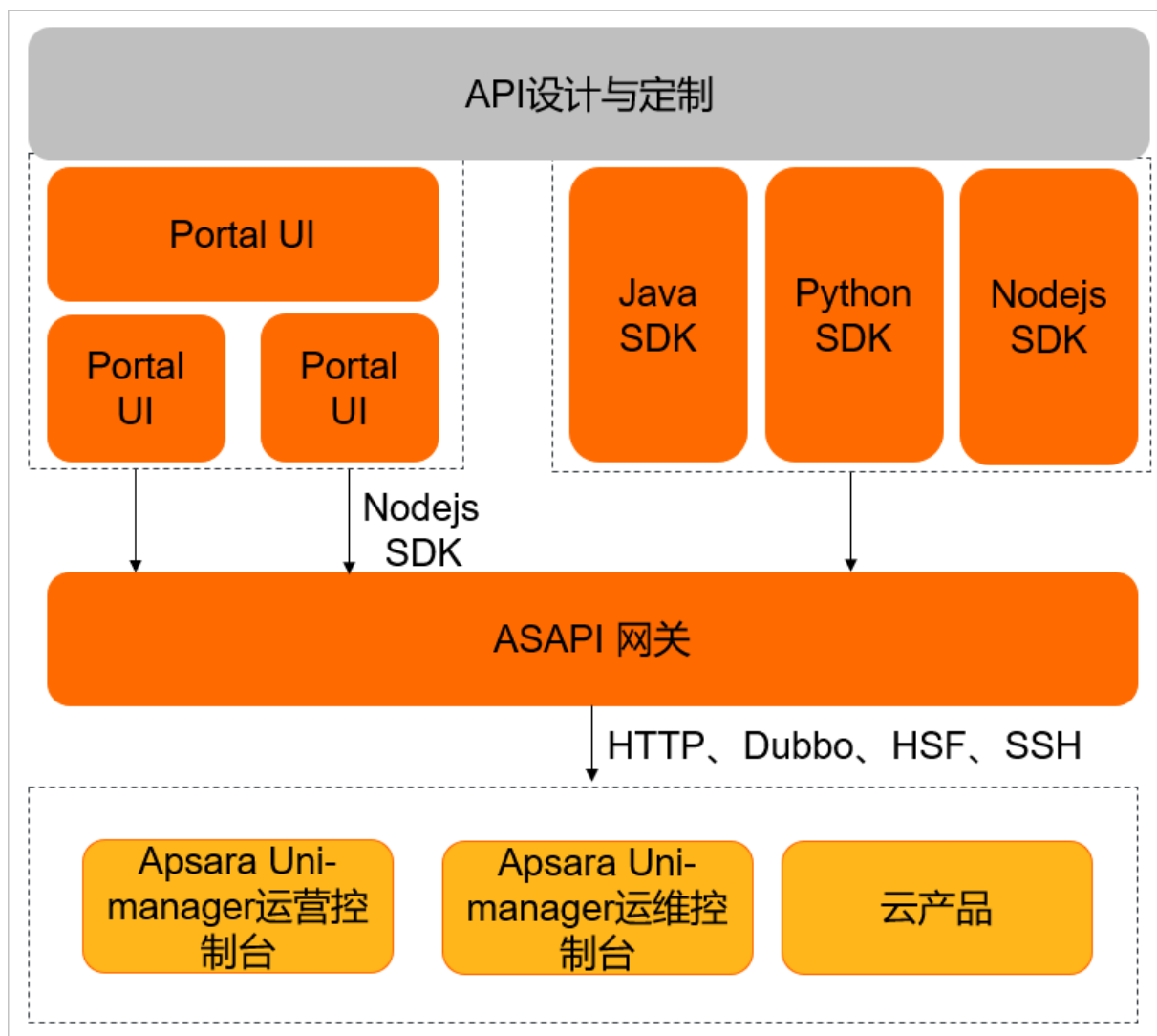
# http的代理
os.environ["HTTP_PROXY"] = "https://x.x.x.x:xxxx"
```

1.3. 通过ASAPI网关调用专有云API

1.3.1. ASAPI概述

本文主要从功能和架构等方面对专有云ASAPI（Apsara Stack API）系统进行简单的介绍。

ASAPI是阿里云专有云的统一入口网关，是阿里云混合云管理平台（Apsara Uni-manager）的核心组成部分。ASAPI定位于专有云API的统一管理和调用，承接了ECS、VPC等绝大多数专有云产品的接入，对外呈现统一的调用方式（统一SDK方式，可调用所有支持的专有云产品）。



ASAPI提供混合云管理相关的运营与运维API接口，包括：

- 组织、用户、角色、资源集、登录策略等企业管理API（相关功能由Apsara Uni-manager运营控制台提供）。
- 配额、计量、规格、密码策略等运营管理API（相关功能由Apsara Uni-manager运营控制台提供）。
- 云产品相关创建、删除、查询、修改等资源管理API（例如创建ECS、创建VPC，相关功能由ECS、VPC等云产品提供）。
- 云产品监控指标查询API（相关功能由Apsara Uni-manager运营控制台提供）。
- 监控告警、库存管理、任务管理、物理平台等运维相关API（相关功能由Apsara Uni-manager运维控制台提供）。

提供)。

上述API可以查询相应的开发指南，例如，组织用户相关API可查询*Apsara Uni-manager运营控制台开发指南*，创建ECS的API则需要查询*云服务器ECS开发指南*。

 **说明** 云产品的开发指南中，如果包含“通过ASAPI调用云产品SDK”，则说明专有云产品已经接入ASAPI，可参照本文进行API调用。

1.3.2. 生成ASAPI的Endpoint

本文将为您介绍三种获取Endpoint的方法。

方法一：通过服务注册变量获取Endpoint

1. 登录飞天基础运维平台。
 - i. 登录Apsara Uni-manager运维控制台。
 - ii. 在顶部菜单栏单击产品运维。
 - iii. 单击云平台运维下方的飞天基础运维平台。
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：通过资源申请报表获取Endpoint

1. 登录飞天基础运维平台。
 - i. 登录Apsara Uni-manager运维控制台。
 - ii. 在顶部菜单栏单击**产品运维**。
 - iii. 单击云平台运维下方的**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. 单击操作栏中的**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` 。
 - 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.████████████████████.g.com",
  "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

③ 说明

1. region: 替换为真实的地域ID值，获取方式请参见[获取Region ID](#)。
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

1.3.3. 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。

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。

1.3.4. 公共参数

公共参数是所有接口调用需要的参数。

公共请求参数

以下是调用ASAPI服务的公共请求参数：

名称	类型	是否必需	描述
Action	String	是	产品注册在ASAPI的API名称，获取方式请参见 查看API信息 。
Product	String	是	产品名称。
Version	String	是	产品注册在ASAPI的API版本，获取方式请参见 查看API信息 。
RegionId	String	是	Region ID。获取方式请参见 获取Region ID 。
AccessKeyId	String	是	访问服务使用的密钥ID，获取方式请参见 获取AccessKey 。
Signature	String	是	您的签名。取值请参见签名机制。
SignatureMethod	String	是	签名方式。取值：HMAC-SHA1，HMAC-SHA256。
SignatureVersion	String	是	签名算法版本。取值：1.0或2.0。

名称	类型	是否必需	描述
SignatureNonce	String	是	签名唯一随机数。用于防止网络重放攻击，建议您每一次请求都使用不同的随机数。
Timestamp	String	是	请求的时间戳。按照ISO8601标准表示，并需要使用UTC时间，格式为yyyy-MM-ddTHH:mm:ssZ。示例：2018-01-01T12:00:00Z表示北京时间2018年01月01日20点00分00秒。
Format	String	是	返回参数的语言类型。取值范围：JSON。

? 说明

其中 `Signature`、`SignatureMethod`、`SignatureVersion`、`SignatureNonce`、`Timestamp`、`Format` 六个参数已经封装在SDK内部，不需要提供。

1.3.5. SDK说明

1.3.5.1. Java SDK调用

本文介绍如何获取Java SDK并开始调用。

前提条件

- 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。在Apsara Uni-manager运营控制台上获取和查看AccessKey的方法请参见[获取AccessKey](#)。
- 获取ASAPI的Endpoint，具体操作请参见[生成ASAPI的Endpoint](#)。
- 阿里云专有云Java SDK需要1.8及以上版本的JDK。

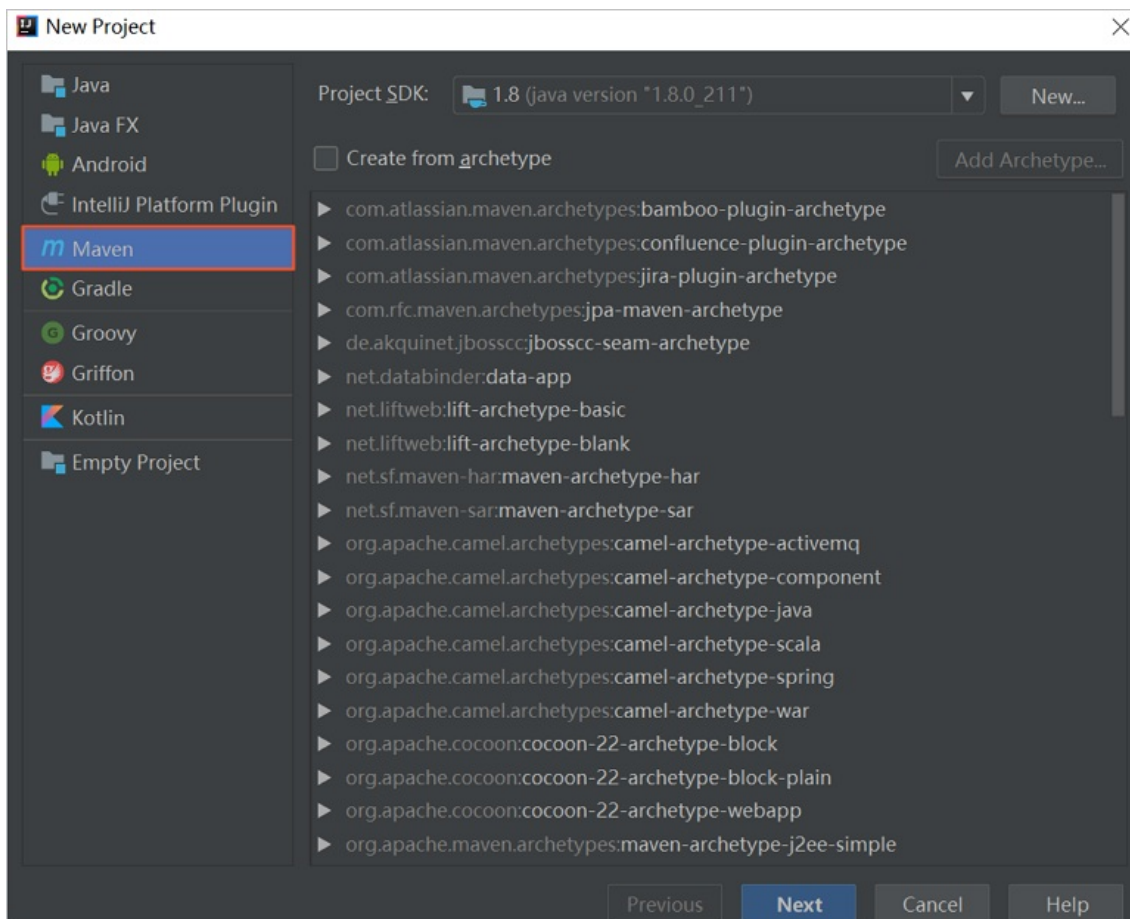
背景信息

专有云帮助中心为您提供了ASAPI Java SDK的Maven项目依赖及JAR包，您可以编写代码调用ASAPI Java SDK来实现访问专有云网关服务ASAPI。

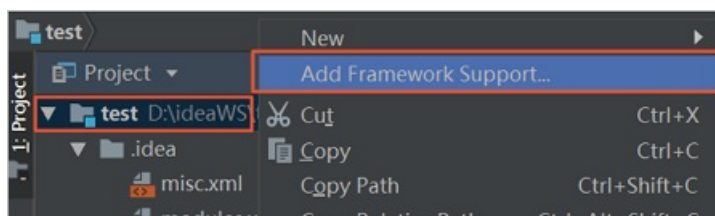
本文示例中，客户端为Windows10 64位操作系统，使用的Java开发工具为IntelliJ IDEA。

操作步骤

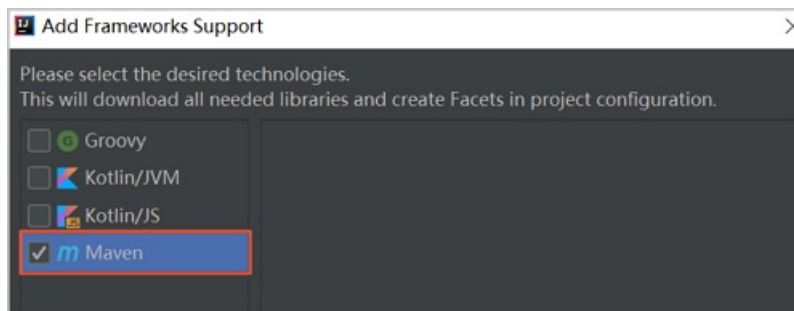
1. 通过以下任一方式在IDEA中配置Maven项目管理工具。
 - 使用IDEA中集成的Maven项目管理工具。
 - 访问Maven官方下载页面（[Download Apache Maven](#)）下载对应操作系统的Maven工具，手动配置Maven工具。
2. 从共享资源中获取ASAPI Java SDK的Maven项目依赖，获取方式请参见[获取ASAPI的SDK包](#)。
3. 通过以下任一方式创建Maven项目。
 - 方式一：在IDEA中添加一个Maven项目。



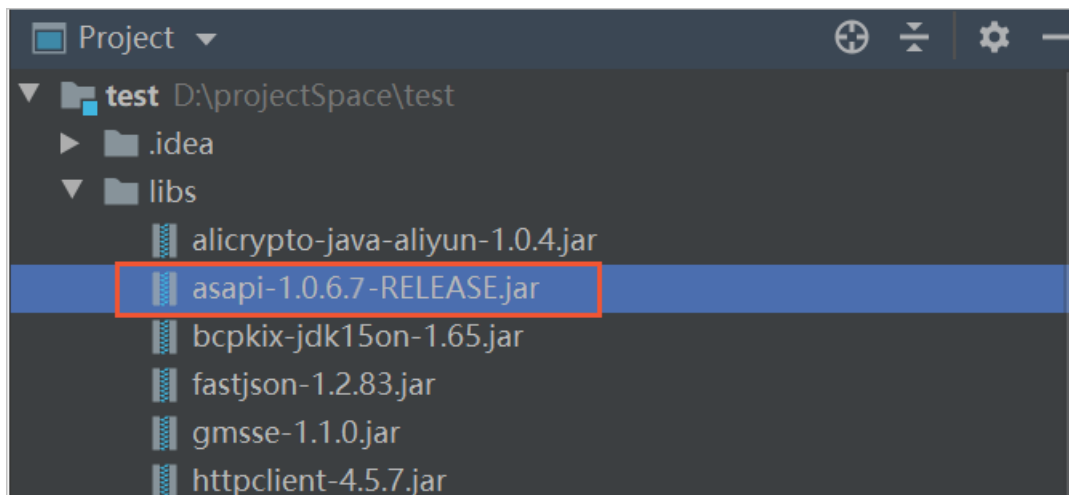
- 方式二：将已有的项目转换为Maven项目。
 - a. 右键单击要转换的项目，并选择**Add Framework Support...**。



b. 选择Maven，并单击OK。



4. 在项目根目录下创建libs包，将ASAPI Java SDK包放入到libs下，如下图。



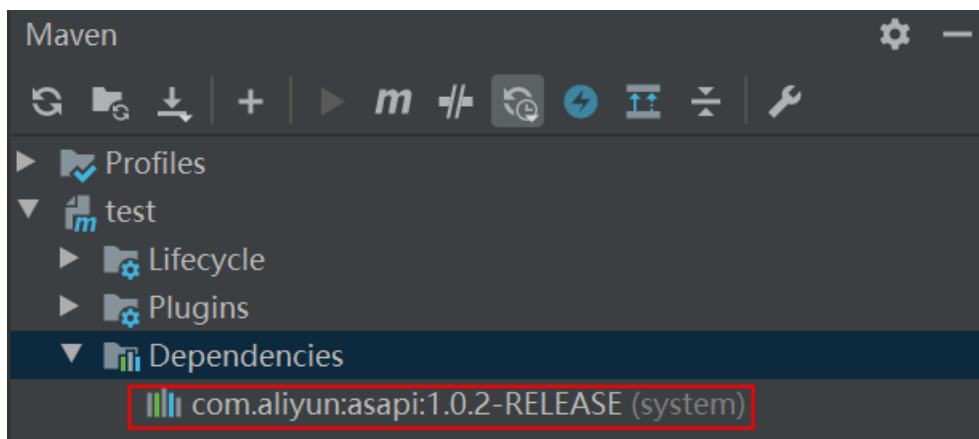
5. 在项目目录下的pom.xml中，添加ASAPI Java SDK依赖。

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>asapi</artifactId>
    <version>1.0.6.4-RELEASE</version>
    <scope>system</scope>
    <systemPath>${project.basedir}/libs/asapi-1.0.6.4-RELEASE.jar</systemPath>
  </dependency>
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>alicrypto-java-aliyun</artifactId>
    <version>1.0.4</version>
    <systemPath>${project.basedir}/libs/alicrypto-java-aliyun-1.0.4.jar</systemPath>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>gmsse</artifactId>
    <version>1.1.0</version>
    <systemPath>${project.basedir}/libs/gmsse-1.1.0.jar</systemPath>
  </dependency>
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.7</version>
    <systemPath>${project.basedir}/libs/httpclient-4.5.7.jar</systemPath>
  </dependency>
</dependencies>
```

❓ 说明 代码中的参数 1.0.6.4-RELEASE 需要根据实际SDK包版本号进行编写。

6. 查看Idea右侧Maven栏。

如图所示，表示已成功导入依赖。



使用Java SDK

目前支持使用GET或POST中任意一种方式调用ASAPI平台的API。调用示例如下：

```
// 1. 创建ASClient连接
ASClient client = new ASClient();
// 2. 设置身份标识,主要是标识调用来源,无实际作用,可随意设置,必填项
client.setSdkSource("<调用身份标识>");
Map<String, Object> parameters = new HashMap<String, Object>();
// 3. 设置授权相关参数 (与环境相关)
// 授权账号的AccessKey ID
parameters.put("AccessKeyId", "<AccessKey ID>");
// 授权账号的AccessKey Secret
parameters.put("AccessKeySecret", "<AccessKey Secret>");
// 地域Region
parameters.put("RegionId", "<your RegionID>");
// 4. 设置调用的接口信息,Product/Action/Version
parameters.put("Product", "<产品名称>");
parameters.put("Action", "<接口名称>");
parameters.put("Version", "<接口版本>");
// 当使用sts ak调用rpc接口时 需要在参数中加上SecurityToken字段
// parameters.put("SecurityToken", "<your-securitytoken>");
// 当使用sts ak调用roa接口时 需要在header中加上x-acis-security-token字段
// headers.put("x-acis-security-token", "<your-securitytoken>");
// 5. 接口业务参数设置
parameters.put("<业务参数>", "<参数值>");
// 6. 创建Headers,并设置签名加密方式,目前支持HMAC-SHA1/HMAC-SHA256两种,默认使用HMAC-SHA256
Map<String, String> headers = new HashMap<String, String>();
//headers.put("SignatureMethod", "HMAC-SHA1");
headers.put("x-acis-regionid", "<your-regionid>");
headers.put("x-acis-resourcegroupid", "<your-resourcegroupid>");
headers.put("x-acis-organizationid", "<your-organizationid>");
headers.put("x-acis-instanceid", "<your-instanceid>");
// 7. 发起调用,调用方式推荐使用doPost
// 设置连接超时
//client.setConnectTimeout(10000);
// 设置请求超时
// client.setSocketTimeout(10000);
String result = client.doPost(endpoint, headers, parameters);
System.out.println(result);
```

设置代理

访问代理环境时调用添加代理如下:

```
Cloud cloud = new Cloud();
// 设置IP
cloud.setConfig("proxy_host", "<IP>");
// 设置端口
cloud.setConfig("proxy_port", "<Port>");
// 创建ASClient连接
ASClient client = new ASClient(cloud);
```

Header参数描述

名称	描述
x-acs-regionid	环境Region，获取方式请参见 获取Region ID 。
x-acs-organizationid	Apsara Uni-manager运营控制台上的组织ID。
x-acs-resourcegroupid	Apsara Uni-manager运营控制台上的资源集ID。
x-acs-instanceid	进行操作的实例ID。

1.3.5.2. Python SDK调用

您可以使用ASAPI提供的Python SDK进行API调用。

环境准备

- 用阿里云专有云Python SDK，您需要一个授权账号以及一对AccessKey ID和AccessKey Secret。请参见[获取AccessKey](#)在Apsara Uni-manager运营控制台上获取和查看您的AccessKey。
- 获取ASAPI的Endpoint，具体操作请参见[生成ASAPI的Endpoint](#)。

安装ASAPI Python SDK包

- 下载对应版本的SDK包。具体操作请参见[获取ASAPI的SDK包](#)。
- 将SDK包中Python SDK的压缩包上传至需要运行Python API调用程序的目标主机中并解压，然后在SDK文件包所在目录下执行以下代码安装SDK包：

```
# Python 2.x执行
pip install aliyun-python2-sdk-asapi-2.4.7.tar.gz
# Python 3.x执行
pip3 install aliyun-python3-sdk-asapi-2.4.7.tar.gz
```

? 说明

代码中的 `2.4.7` 需要根据实际使用的SDK包版本号进行编写。

调用示例

目前支持GET和POST两种方式调用ASAPI平台的API。

```
# 1. 引入依赖
from aliyunsdkasapi.ASClient import ASClient
from aliyunsdkasapi.AsapiRequest import AsapiRequest

if __name__ == '__main__':
    # 2. 创建Request, Product产品名称, Version接口版本, Action接口名称, asapi-endpoint ASAPI服务访问
    域名
    req = AsapiRequest("<Product>", "<Version>", "<Action>", "<asapi-endpoint>")
    # 3. 业务信息
    req.add_body_params("<key>", "<value>")
    # 当使用sts ak调用rpc接口时 需要在参数中加上SecurityToken字段
    # req.add_body_params("SecurityToken", "<your-securitytoken>");
    # 当使用sts ak调用roa接口时 需要在header中加上x-acis-security-token字段
    # req.add_header("x-acis-security-token", "<your-securitytoken>");
    # 4. Header信息
    req.add_header("x-acis-resourcegroupid", "<your-resourcegroupid>");
    req.add_header("x-acis-organizationid", "<your-organizationid>");
    req.add_header("x-acis-regionid", "<your-regionid>");
    req.add_header("x-acis-instanceid", "<your-instanceid>");
    # 5. 设置请求方式
    req.set_method("POST")
    # 6. 用环境信息初始化ASClient timeout设置请求超时时间, cert_file设置证书文件, verify是否验证证书
    as_client = ASClient("<accesskeyid>", "<accesskeysecret>", "<regionid>", timeout=1000,
                        cert_file=None,
                        verify=False)
    # 设置调用源, 主要是标识调用来源, 无实际作用, 可随意设置, 必填项
    as_client.setSdkSource("<调用身份标识>")
    response = as_client.do_request(req)
    print(response)
```

Header参数描述

名称	描述
x-acis-resourcegroupid	Apsara Uni-manager运营控制台上的资源集ID。
x-acis-organizationid	Apsara Uni-manager运营控制台上的组织ID。
x-acis-regionid	环境Region，获取方式请参见 获取Region ID 。
x-acis-instanceid	进行操作的实例ID。

设置代理

您需要按以下步骤设置代理：

- 1. 执行以下命令导入OS依赖。


```
import os
```

2. 执行以下命令设置代理。

```
# 调用开始前添加
os.environ['http_proxy'] = "http://x.x.x.x:xxxx"
os.environ['https_proxy'] = "https://x.x.x.x:xxxx"
```

1.3.5.3. Node SDK调用

本文介绍如何获取Node.js SDK并开始调用。

环境准备

- 使用阿里云专有云Node.js SDK，您需要获取授权账号、一对AccessKey ID和AccessKey Secret。具体操作请参见[获取AccessKey](#)。
- 获取ASAPI的Endpoint，具体操作请参见[生成ASAPI的Endpoint](#)。
- 获取Region ID，具体操作请参见[获取Region ID](#)。

添加依赖

执行以下命令安装Node.js SDK包。

```
tnpm i @ali/asapi-nodejs-sdk@0.0.33 --save
```

使用Node.js SDK

以下展示了调用Node.js SDK的示例代码：

```
// 1. 引入调用SDK依赖
const Controller = require('egg').Controller;
const ASAPIClient = require('@ali/asapi-nodejs-sdk');
// 2. 设置配置
const cnofig = {
  API_GATEWAY: '<ASAPI的Endpoint>',
  regionId: '<云产品所在地域>',
  accessKeyId: '<授权账号的AK>',
  accessKeySecret: '<授权账号的SK>'
};
// 3. 初始化ASAPI客户端
const AsapiClient = new ASAPIClient(cnofig, this.ctx);
// 4. 配置请求参数
const params = {
  "Action": "<接口名称>",
  "Product": "<接口所属产品>",
  "Version": "<接口版本>",
  "regionId": "<云产品所在地域>",
  // 当使用sts ak调用rpc接口时 需要在参数中加上SecurityToken字段
  // "SecurityToken": "<your-securitytoken>",
  "<接口所需业务参数>": "<参数值>"
}
// 5. 添加headers
// 当使用sts ak调用roa接口时 需要在header中加上x-acis-security-token字段
// AsapiClient.addHeader("x-acis-security-token", "<your-securitytoken>")
AsapiClient.addHeader('x-acis-caller-sdk-source', '<身份标识>')
AsapiClient.addHeader('x-acis-roleid', '<角色id>')
AsapiClient.addHeader('x-acis-userid', '<用户id>')
AsapiClient.addHeader('x-acis-organizationid', '<your-organizationid>')
AsapiClient.addHeader("x-acis-regionid", "<your-regionid>");
AsapiClient.addHeader("x-acis-resourcegroupid", "<your-resourcegroupid>");
AsapiClient.addHeader("x-acis-instanceid", "<your-instanceid>");
// 6. 发起调用并且获取返回结果
try {
  const res = await AsapiClient.doRequest(cnofig.API_GATEWAY, params, {
    method: 'POST', // 请求方式
    formatParams: false, // 是否格式化参数,首字母大写转化
    enableProxy: true, // 是否配置代理
    proxy: 'http://x.x.x.x:xxxx',
    timeout: 10000, // 超时时间
    // ca: ,// ca证书设置 详见 https://nodejs.org/api/https.html#https_https_request_options_c
  });
  console.log(res)
} catch (e) {
  console.log(e)
}
```

设置忽略证书

以下展示了忽略证书的代码。

设置环境变量

NODE_TLS_REJECT_UNAUTHORIZED=0

Header参数描述

名称	描述
x-acs-regionid	环境Region，获取方式请参见 获取Region ID 。
x-acs-organizationid	Apsara Uni-manager运营控制台上的组织ID。
x-acs-resourcegroupid	Apsara Uni-manager运营控制台上的资源集ID。
x-acs-instanceid	进行操作的实例ID。
x-acs-caller-sdk-source	输入自定义的调用方名称或者系统名称，作为来源说明。
x-acs-roleid	角色的ID。
x-acs-userid	用户的ID。

1.3.6. HTTP调用方式

1.3.6.1. 概述

本文介绍HTTP调用方式，适用基于API URL发起HTTP GET请求的用户。

如果您使用的是SDK可以跳过此环节。

发起API请求的URL由不同参数拼凑而成，有固定的请求结构。URL中包含公共参数、您的签名和某个API的具体参数。每篇API文档均给出了URL请求示例供您参考，但是为了方便显示，我们并没有编码这些URL示例，您需要在发起请求前自行编码。我们根据您的签名验证了请求后，会返回结果给您。接口调用成功会显示返回参数，调用失败则显示相应报错，您可以根据错误信息分析排查。

1.3.6.2. 请求结构

专有云ASAPI支持基于URL发起HTTP GET请求。请求参数需要包含在URL中。本文列举了GET请求中的结构解释，并提供了ASAPI的服务接入地址（Endpoint）。

结构示例

以下为GetOrganizationList一条未编码的URL请求示例：

```
https://public.asapi.aliyuns.com/asapi/v3?Action=GetOrganizationList&Product=ascm&Version=2019-05-10&Id=<组织Id>&<公共请求参数>
```

- `https` 指定了请求通信协议。
- `public.asapi.aliyuns.com/asapi/v3` 指定了ASAPI的服务接入地址（Endpoint）。

- `Action=GetOrganizationList Product=ascm Version=2019-05-10` 指定了要调用的API信息。
- `Id` 是组织的ID。
- `<公共请求参数>` 是系统规定的公共参数。

通信协议

ASAPI只支持HTTPS协议请求通信。

接入地址

ASAPI服务接入地址，请参照[生成ASAPI的Endpoint](#)。

您需要通过 `Action/Product/Version` 参数指定目标操作，例如

`Action=GetOrganizationList Product=ascm Version=2019-05-10`。还需要指定接口的其他参数以及公共请求参数，请参见[公共参数](#)。

1.3.6.3. 签名机制

对于每一次HTTP调用，我们会根据访问中的签名信息验证访问请求者身份。具体由使用AccessKey ID和AccessKey Secret对称加密验证实现。

② 说明

- Apsara Uni-manager运营控制台对每个用户和组织都生成了对应的AccessKey ID和AccessKey Secret。
- AccessKey ID是访问者身份，AccessKey Secret是加密签名字符串和服务器端验证签名字符串的密钥，必须严格保密谨防泄露。
- ASAPI目前提供Node.js、Python和Java三种语言的SDK，可以免去您签名的烦恼。更多详情，请参照[获取ASAPI的SDK包](#)。

步骤一：构造规范化请求字符串

1. 排序参数。排序规则以首字母顺序排序，排序参数包括公共请求参数和接口自定义参数，不包括公共请求参数中的 `Signature` 参数。

② 说明

当使用GET方法提交请求时，这些参数就是请求URL中的参数部分，即URL中问号（?）之后由与号（&）连接的部分。

2. 编码参数。使用UTF-8字符集按照RFC3986规则编码请求参数和参数取值，编码规则如下：
 - 字符英文大写或小写字母（A~Z、a~z）、阿拉伯数字（0~9）以及字符短划线（`-`）、下划线（`_`）、英文句号（`.`）、不编码。

- 其它字符编码成 `%XY` 的格式，其中 `XY` 是字符对应ASCII码的16进制。示例：半角双引号（`"`）对应 `%22`。
- 扩展的UTF-8字符，编码成 `%XY%ZA...` 的格式。
- 空格（）编码成 `%20`，而不是加号（`+`）。

该编码方式与 `application/x-www-form-urlencoded` MIME格式编码算法相似，但又有所不同。

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

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws UnsupportedOperationException
{
    return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace(
        "*", "%2A").replace("%7E", "~") : null;
}
```

- 使用等号（`=`）连接编码后的请求参数和参数取值。
- 使用与号（`&`）连接编码后的请求参数，注意参数排序与步骤1一致。

现在，您得到了规范化请求字符串（`CanonicalizedQueryString`），其结构遵循[请求结构](#)。

步骤二：构造签名字符串

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

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

- 按照[RFC2104](#)的定义，计算待签名字符串 `StringToSign` 的HMAC-SHA1或者HMAC-256值。示例使用的是Java Base64编码方法。

```
String signature = Base64.getEncoder().encodeToString(signData);
```

❓ 说明

计算签名时，RFC2104规定的Key值是您的 `AccessKeySecret` 并加上与号（`&`），其ASCII值为38。
更多详情请参见[获取AccessKey](#)。

3. 添加根据RFC3986规则编码后的参数 `Signature` 到规范化请求字符串URL中。

示例一：参数拼接法

以调用 `GetOrganizationList` 查询地域为例。假设您获得了 `AccessKeyId=testid` 以及 `AccessKeySecret=testsecret`，签名流程如下所示：

1. 构造规范化请求字符串。

```
https://public.asapi.aliyuns.com/asapi/v3/?AccessKeyId=testid&Action=GetOrganizationList&Format=JSON&Id=1&Product=ascm&RegionId=cn-region-test&SignatureMethod=HMAC-SHA1&SignatureNonce=43967c6e-e3ad-44fa-b06b-fcf3db5f39d2&SignatureVersion=1.0&Timestamp=2020-09-15T08%3A50%3A40Z&Version=2019-05-10
```

2. 构造待签名字符串 `StringToSign`。

```
GET&%2F&AccessKeyId%3Dtestid%26Action%3DGetOrganizationList%26Format%3DJSON%26Id%3D1%26Product%3Dascm%26RegionId%3Dcn-region-test%26SignatureMethod%3DHMAC-SHA1%26SignatureNonce%3D43967c6e-e3ad-44fa-b06b-fcf3db5f39d2%26SignatureVersion%3D1.0%26Timestamp%3D2020-09-15T08%253A50%253A40Z%26Version%3D2019-05-10
```

3. 计算签名值。

计算签名值。因为 `AccessKeySecret=testsecret`，用于计算的Key为 `testsecret&`，计算得到的签名值为 `OLeaidSlJvxuMvnyHOwuJ+uX5qY=`。示例使用的是JavaBase64编码方法。

```
String signature = Base64.getEncoder().encodeToString(signData);
```

 添加RFC3986规则编码后的 `Signature=hvqv9Ll5RE6kQ4YsQXFaf9VuRIY=` 到步骤1的URL中。

```
https://public.asapi.aliyuns.com/asapi/v3/?AccessKeyId=testid&Action=GetOrganizationList&Format=JSON&Id=1&Product=ascm&RegionId=cn-region-test&SignatureMethod=HMAC-SHA1&SignatureNonce=43967c6e-e3ad-44fa-b06b-fcf3db5f39d2&SignatureVersion=1.0&Timestamp=2020-09-15T08%3A50%3A40Z&Version=2019-05-10
```

通过以上URL，您可以使用浏览器、curl或者wget等工具发起HTTP请求调用 `DescribeRegions`，查看阿里云的地域列表。

示例二：编程语言法

依然以调用 `GetOrganizationList` 查询地域为例。假设您获得了 `AccessKeyId=testid` 以及 `AccessKeySecret=testsecret`，并且假定所有请求参数放在一个Java `Map<String, String>` 对象里。

1. 预定义编码方法。

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws UnsupportedEncodingException {
    return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("*", "%2A").replace("%7E", "~") : null;
}
```

2. 预定义编码时间格式 `Timestamp`。参数 `Timestamp` 必须符合ISO8601规范，并需要使用UTC时间，时区为+0。

```
private static final String ISO8601_DATE_FORMAT = "yyyy-MM-dd'T'HH:mm:ss'Z'";
private static String formatIso8601Date(Date date) {
    SimpleDateFormat df = new SimpleDateFormat(ISO8601_DATE_FORMAT);
    df.setTimeZone(new SimpleTimeZone(0, "GMT"));
    return df.format(date);
}
```

3. 构造请求字符串。

```
final String HTTP_METHOD = "GET";
Map parameters = new HashMap();
// 输入请求参数
parameters.put("Product", "ascm");
parameters.put("Action", "GetOrganizationList");
parameters.put("Version", "2019-05-10");
parameters.put("AccessKeyId", "testid");
parameters.put("RegionId", "cn-region-test");
parameters.put("Timestamp", formatIso8601Date(new Date()));
parameters.put("SignatureMethod", "HMAC-SHA1");
parameters.put("SignatureVersion", "1.0");
parameters.put("SignatureNonce", UUID.randomUUID().toString());
parameters.put("Format", "JSON");
parameters.put("Id", "1");
// 排序请求参数
String[] sortedKeys = parameters.keySet().toArray(new String[]{});
Arrays.sort(sortedKeys);
final String SEPARATOR = "&";
// 构造 stringToSign 字符串
StringBuilder stringToSign = new StringBuilder();
stringToSign.append(HTTP_METHOD).append(SEPARATOR);
stringToSign.append(percentEncode("/").append(SEPARATOR));
StringBuilder canonicalizedQueryString = new StringBuilder();
for(String key : sortedKeys) {
    // 这里注意编码 key 和 value
    canonicalizedQueryString.append("&")
        .append(percentEncode(key)).append("=")
        .append(percentEncode(parameters.get(key)));
}
// 这里注意编码 canonicalizedQueryString
stringToSign.append(percentEncode(
    canonicalizedQueryString.toString().substring(1)));
```

4. 签名。因为 `AccessKeySecret=testsecret`，所以用于计算HMAC的Key为 `testsecret&`，计算得到的签名为 `OLeaidS1JvXuMvnyH0wuJ%2BuX5qY%3D`。

```
// 以下是一段计算签名的示例代码
final String ALGORITHM = "HmacSHA1";
final String ENCODING = "UTF-8";
key = "testsecret&";
Mac mac = Mac.getInstance(ALGORITHM);
mac.init(new SecretKeySpec(key.getBytes(ENCODING), ALGORITHM));
byte[] signData = mac.doFinal(stringToSign.getBytes(ENCODING));
String signature = Base64.getEncoder().encodeToString(signData);
```

增加签名参数后，按照[RFC3986](#)规则编码后的URL如下所示：

```
https://public.asapi.aliyuns.com/asapi/v3/?AccessKeyId=testid&Action=GetOrganizationList&Format=JSON&Id=1&Product=ascm&RegionId=cn-region-test&SignatureMethod=HMAC-SHA1&SignatureNonce=43967c6e-e3ad-44fa-b06b-fcf3db5f39d2&SignatureVersion=1.0&Timestamp=2020-09-15T08%3A50%3A40Z&Version=2019-05-10
```

5. 使用浏览器、curl或者wget等工具发送HTTP请求。

1.3.6.4. 返回结果

返回结果目前只支持JSON结果的输出。当返回码大于等于200且小于300时为正常返回，否则为异常返回。

正常返回示例

接口调用成功后一般会返回请求ID和具体业务返回值，具体可参见云产品开发手册说明。


```
{
  "successResponse":true,
  "eagleEyeTraceId":"xxxxxxxxxxxxxxxx",
  "asapiSuccess":true,
  "code":"200",
  "cost":12,
  "asapiRequestId":"xxxxxxxxxxxxxxxx",
  "data":[
    {
      "muserId":"aliyuntest",
      "internal":false,
      "level":"0",
      "supportRegions":"cn-*",
      "personNum":100,
      "mtime":1553247962000,
      "uuid":"1",
      "parentId":0,
      "supportRegionList":[
        "cn-*"
      ],
      "name":"root",
      "alias":"root",
      "ctime":1553247962000,
      "id":1,
      "resourceGroupNum":100,
      "cuserId":"aliyuntest"
    }
  ],
  "requestId":"xxxxxxxxxxxxxxxx",
  "success":true,
  "message":"success"
}
```

异常返回示例

接口调用出错后，会返回如下结构，具体可参照云产品开发手册说明。

```
{
  "code": "asapi.server.api.notfound", //错误代码
  "eagleEyeTraceId": "xxx-xxx-xxx-xxx", //鹰眼
  "asapiSuccess": false, //本次请求是否成功
  "hostId": "xxxx", //主机ID
  "message": "Failed to ...", //消息
  "asapiErrorMessage": "Failed to ...", //ASAPI错误信息
  "asapiErrorCode": "asapi.server.api.notfound" //ASAPI错误码
}
```

2.API参考

2.1. API 概览

实例生命周期管理接口

API	描述
CreateInstance	创建实例
DeleteInstance	释放实例
ModifyInstanceSpec	变更实例

实例管理接口

API	描述
DescribeInstanceAttribute	查询实例详情
DescribeInstances	查询实例基础信息
DescribeRegions	查询实例可售卖地域
DescribeSecurityIps	查询允许访问实例的 IP 名单
FlushInstance	清空实例
ModifyInstanceAttribute	修改实例属性
ModifyInstanceMaintainTime	修改实例可运维时间
ModifySecurityIps	修改实例白名单
ModifyInstanceNetExpireTime	修改经典网络域名的保留时间
SwitchNetwork	修改网络类型

备份恢复接口

API	描述
CreateBackup	创建备份
ModifyBackupPolicy	修改备份策略
DescribeBackupPolicy	查询备份策略
DescribeBackups	显示备份列表
RestoreInstance	基于备份集回滚实例

监控管理接口

API	描述
DescribeMonitorItems	查看监控项列表
DescribeHistoryMonitorValues	查看历史监控

参数管理接口


API	描述
DescribeInstanceConfig	查看实例参数配置
ModifyInstanceConfig	修改参数配置

2.2. 请求结构

服务地址

Redis API 的服务接入地址为 `r-kvstore.example.com`。

接入地址的获取方法，请参见[生成ASAPI的Endpoint](#)。

 说明 `example.com`是项目自定义的访问域名，请联系管理员获取该域名。

通信协议

支持通过 HTTP 或 HTTPS 通道进行请求通信。为了获得更高的安全性，推荐您使用 HTTPS 通道发送请求。

请求方法

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

请求参数

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

字符编码

请求及返回结果都使用 UTF-8 字符集进行编码。

2.3. 公共参数

公共请求参数是指每个接口都需要使用到的请求参数。

公共请求参数

公共请求参数

名称	类型	是否必须	描述
Format	String	是	返回值的类型，支持 JSON 与 XML。默认为 XML。
Version	String	是	API 版本号，为日期形式：YYYY-MM-DD，本版本对应为 2015-01-01。
AccessKeyId	String	是	阿里云颁发给用户的访问服务所用的密钥 ID。
Signature	String	是	签名结果串，关于签名的计算方法，请参见 签名机制 。
SignatureMethod	String	是	签名方式，目前支持 HMAC-SHA1。
Timestamp	String	是	请求的时间戳。日期格式按照ISO8601标准表示，并需要使用 UTC 时间。格式为 YYYY-MM-DDThh:mm:ssZ。例如，2016-01-01T12:00:00Z（为北京时间2016年1月1日20点0分0秒）。
SignatureVersion	String	是	签名算法版本，目前版本是1.0。
SignatureNonce	String	是	唯一随机数，用于防止网络重放攻击。用户在不同请求时间要使用不同的随机数值。

```
https:// r-kvstore.example.com/
?Format=json
&Version=2015-01-01
&Signature=Pc5xxxxxxxxxxxxxxxxxxxx
&SignatureMethod=HMAC-SHA1
&SignatureNonce=15215528852396
&SignatureVersion=1.0
&AccessKeyId=key-test
&Timestamp=2016-01-01T12:00:00Z
```


公共返回参数

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

```
<?xml version="1.0" encoding="utf-8"?>
<!--结果的根结点-->
<接口名称+Response>
<!--返回请求标签-->
<RequestId>4C4fdsf38-3910-447D-87AD-AC078932216</RequestId>
<!--返回结果数据-->
</接口名称+Response>
```

2.4. 返回结果

调用 API 服务后返回数据采用统一格式。返回的 HTTP 状态码为 2xx 代表调用成功；返回 4xx 或 5xx 的 HTTP 状态码代表调用失败。调用成功返回的数据格式主要有 XML 和 JSON 两种，外部系统可以在请求时传入参数来制定返回的数据格式，默认为 XML 格式。

 **说明** 本文档中的返回示例为了便于用户查看，做了格式化处理，实际返回结果是没有进行换行、缩进等处理的。

成功结果

XML 返回结果包括请求是否成功信息和具体的业务数据。示例如下：

```
<?xml version="1.0" encoding="utf-8"?>
<!--结果的根结点-->
<接口名称+Response>
<!--返回请求标签-->
<RequestId>4C4fdsf38-3910-447D-87AD-AC078932216</RequestId>
<!--返回结果数据-->
</接口名称+Response>
```

JSON 示例：

```
{
  "RequestId": "4C4fdsf38-3910-447D-87AD-AC078932216",
  /* 返回结果数据 */
}
```

错误结果

调用接口出错后，将不会返回结果数据。调用方可根据附表错误代码表来定位错误原因。当调用出错时，HTTP 请求返回一个 4xx 或 5xx 的 HTTP 状态码。返回的消息体中是具体的错误代码及错误信息。另外还包含一个全局唯一的请求 ID:RequestId 和一个您该次请求访问的站点 ID:HostId。若在调用方找不到错误原因，可以联系专有云客服，并提供该 HostId 和 RequestId，以便我们尽快帮您解决问题。

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
<RequestId>8906582E-6722-409A-A6C4-0E7863B733A5</RequestId>
<HostId> r-kvstore.example.com</HostId>
<Code>UnsupportedOperation</Code>
<Message>The specified action is not supported.</Message>
</Error>
```

JSON 示例：

```
{
  "RequestId": "7463B73D-35CC-4D19-A010-6B8D65D242EF",
  "HostId": " r-kvstore.example.com",
  "Code": "UnsupportedOperation",
  "Message": "The specified action is not supported."
}
```

2.5. 签名机制

Redis 服务会对每个访问的请求进行身份验证，所以无论使用 HTTP 还是 HTTPS 协议提交请求，都需要在请求中包含签名（Signature）信息。Redis 服务通过使用 Access Key ID 和 Access Key Secret 进行对称加密的方法来验证请求的发送者身份。

Access Key ID 和 Access Key Secret 由阿里云官方颁发给访问者（可以通过阿里云官方网站申请和管理），其中 Access Key ID 用于标识访问者的身份；Access Key Secret 是用于加密签名字符串和服务器端验证签名字符串的密钥，必须严格保密，只有阿里云和用户知道。

AccessKey的获取方法，请参见[获取AccessKey](#)。

签名处理

用户在访问时，按照下面的方法对请求进行签名处理：

1. 使用请求参数构造规范化的请求字符串（Canonicalized Query String）。
 - i. 按照参数名称的字典顺序对请求中所有的请求参数（包括文档中描述的“公共请求参数”和指定请求接口的自定义参数，但不能包括“公共请求参数”中提到 Signature 参数本身）进行排序。

② 说明 当使用 GET 方法提交请求时，这些参数就是请求 URI 中的参数部分（即 URI 中“?”之后由“&”连接的部分）。

- ii. 对每个请求参数的名称和值进行编码。

名称和值要使用 UTF-8 字符集进行 URL 编码。URL 编码的编码规则如下：

- 对于字符 A~Z、a~z、0~9 以及字符“-”、“_”、“.”、“~”不编码；
- 对于其它字符编码成 %XY 的格式，其中 XY 是字符对应 ASCII 码的 16 进制表示。比如英文的双引号（"）对应的编码为 %22；
- 对于扩展的 UTF-8 字符，编码成 %XY%ZA... 的格式；
- 英文空格（ ）要编码成 %20，而不是加号（+）。

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

- iii. 对编码后的参数名称和值使用英文等号（=）进行连接。
 - iv. 再把英文等号连接得到的字符串按参数名称的字典顺序依次使用&符号连接，即得到规范化请求字符串。
2. 使用上一步构造的规范化字符串按照下面的规则构造用于计算签名的字符串：


```
StringToSign=
HTTPMethod + "&" +
percentEncode("/") + "&" +
percentEncode(CanonicalizedQueryString)
```

其中：

- HTTPMethod 是提交请求用的 HTTP 方法，比如 GET。


- `percentEncode("/")` 是按照步骤 1.b 中描述的 URL 编码规则对字符 “/” 进行编码得到的值，即 `%2F`。
- `percentEncode(CanonicalizedQueryString)` 是对步骤 1 中构造的规范化请求字符串按步骤 1.b 中描述的 URL 编码规则编码后得到的字符串。

3. 按照 RFC2104 的定义，使用上面的用于签名的字符串计算签名 HMAC 值。

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

4. 按照 Base64 编码规则把上面的 HMAC 值编码成字符串，即得到签名值 (Signature)。

5. 将得到的签名值作为 Signature 参数添加到请求参数中，即完成对请求签名的过程。

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

示例

以 DescribeDBInstances 为例，签名前的请求 URL 如下：

```
http://r-kvstore.example.com/?Timestamp=2013-06-01T10:33:56Z&Format=XML&AccessKeyId=testid&Action=DescribeInstances&SignatureMethod=HMAC-SHA1&RegionId=region1&SignatureNonce=NwDaxvLU6tFE0DVb&Version=2015-01-01&SignatureVersion=1.0
```

计算得到的待签名字符串 StringToSign 为：

```
GET&%2F&AccessKeyId%3Dtestid&Action%3DDescribeInstances&Format%3DXML&RegionId%3Dregion1&SignatureMethod%3DHMAC-SHA1&SignatureNonce%3DNwDaxvLU6tFE0DVb&SignatureVersion%3D1.0&Timestamp%3D2013-06-01T10:33:56Z&Version%3D2015-01-01
```

假如使用的 Access Key Id 是 `testid`，Access Key Secret 是 `testsecret`，用于计算 HMAC 的 Key 就是 `testsecret&`，则计算得到的签名值是 `BIxxxxxxxxxxxxxxxxxxxxxx=`。

签名后的请求 URL 为（注意增加了 Signature 参数）：


```
http://r-kvstore.example.com/?Timestamp=2013-06-01T10:33:56Z&Format=XML&AccessKeyId=testid&Action=DescribeInstances&SignatureMethod=HMAC-SHA1&RegionId=region1&SignatureNonce=NwDaxvLU6tFE0DVb&SignatureVersion=1.0&Version=2015-01-01&Signature=BIPOMLxxxxxxxxxxxxxxxxxLXBeZtLQkJTw6iFvw1E%3D
```

2.6. 如何使用 RAM 授权

2.6.1. 使用 RAM 实现资源授权

您通过云账号创建的实例，都是该账号拥有的资源。默认情况下，账号对自己的资源拥有完整的操作权限。

使用阿里云的 RAM (Resource Access Management) 服务，您可以将云账号下的资源的访问及管理权限授予 RAM 中子用户。

 **说明** 如果您不需要使用 RAM，请略过此章节。

2.6.2. RAM 中可授权的资源类型

云数据库KVStore for Redis可以在 RAM 中进行授权的资源类型仅一种：instance。

在通过 RAM 进行授权时，资源的描述方式如下：

资源类型	授权策略中的资源描述方式
Instance	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid acs:kvstore:\$regionid:\$accountid:instance/ acs:kvstore:::instance/

其中，所有 \$regionid 应为某个 region 的 ID，或者 *；所有 \$instanceid 应为某个 instance 的 ID，或者 *；以此类推，\$account-id 是云账号的数字 ID，可以用 * 代替。

2.6.3. RAM 中可对资源进行授权的 Action

RAM 中可对资源进行授权的 Action 如下列表所示。

- CreateInstance
- DeleteInstance
- ModifyInstanceSpec
- RenewInstance
- RenewMultiInstance
- TransformToPrePaid
- ModifyInstanceAttribute
- FlushInstance
- DescribeInstances
- DescribeInstanceAttribute
- ModifyInstanceMaintainTime
- ModifySecurityIps
- SwitchNetwork
- ModifyInstanceNetExpireTime
- CreateBackup
- ModifyBackupPolicy
- DescribeBackupPolicy
- DescribeBackups
- RestoreInstance
- DescribeHistoryMonitorValues
- DescribeInstanceConfig
- ModifyInstanceConfig

2.6.4. API 的鉴权规则

当子用户通过 API 进行资源访问时，后台向 RAM 进行权限检查，以确保调用者拥有响应权限。

每个不同的 API 会根据涉及到的资源以及 API 的语义来确定需要检查哪些资源的权限。具体地，每个 API 的鉴权规则见下表。

鉴权规则

Action	鉴权规则
CreateDBInstance	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
DeleteInstance	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
ModifyInstanceSpec	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
RenewInstance	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
RenewMultiInstance	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
TransformToPrePaid	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
ModifyInstanceAttribute	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
FlushInstance	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
DescribeInstances	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
DescribeInstanceAttribute	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
ModifyInstanceMaintainTime	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
ModifySecurityIps	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
SwitchNetwork	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
ModifyInstanceNetExpireTime	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid

Action	鉴权规则
CreateBackup	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
ModifyBackupPolicy	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
DescribeBackupPolicy	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
DescribeBackups	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
RestoreInstance	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
DescribeHistoryMonitorValues	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
DescribeInstanceConfig	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid
ModifyInstanceConfig	acs:kvstore:\$regionid:\$accountid:instance/\$instanceid

2.7. 生命周期管理

2.7.1. CreateInstance

调用CreateInstance接口创建一个Redis实例。

请求参数

请求参数


名称	类型	是否必须	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	CreateInstance	操作接口名，取值：CreateInstance。
InstanceName	String	否	apitest	实例名称。名称为2~128个字符，以大小写英文字母或中文开头，不支持空格和特殊字符 @ / : = " < > { [] } 。

名称	类型	是否必须	示例值	描述
Password	String	否	Pass!123456	实例密码。字符类型，8~30个字符，需同时包含大写字母、小写字母和数字。 <div> ? 说明 暂时不支持 <code>!<>()</code> <code>[]{} , ~ . - _ @ # \$ % ^ & *</code> 之类的特殊符号。 </div>
InstanceClass	String	是	redis.master.small.default	Redis实例规格，更多信息，请参见 实例规格表 。
RegionId	String	是	cn-hangzhou	申请的存储实例所在区域，可调用DescribeRegions获取可用的数据中心。
ZoneId	String	否	cn-hangzhou-e	RegionId 下的可用区，可调用DescribeRegions获取可用的可用区。
Token	String	否	ChargeType	用于保证请求的幂等性。由客户端生成该参数值，要保证在不同请求间唯一，大小写敏感、不超过64个ASCII字符。
NetworkType	String	否	VPC	网络类型： <ul style="list-style-type: none"> CLASSIC：经典网络，默认为经典网络。 VPC：专有网络
VpcId	String	否	vpc-bp1nme44gek34slfc****	VPC 网络 ID。
VSwitchId	String	否	vsw-bp1e7clcw529l773d****	虚拟交换机 ID。 <div> ? 说明 如果传入了VpcId参数，则本参数必须传入。 </div>
PrivateIp	String	否	172.16.0.***	私有 IP 地址。
SrcDBInstanceId	String	否	r-bp1zxszhcgatnx****	基于某个实例的备份集创建实例有效，输入产生备份集的实例 ID。
BackupId	String	否	111111111	基于某个实例的备份集创建实例有效，输入产生备份集 ID。可调用DescribeBackups接口获取。
InstanceType	String	否	Redis	实例类型，取值： Redis，Memcache。默认为 Redis。

返回参数

返回参数

名称	参数类型	示例值	描述
<公共返回参数>	无	无	参见 公共返回参数 。
InstanceId	String	r-bp1zxszhcgatnx****	实例ID（全局唯一）。
InstanceName	String	apitest	实例名称。

 **说明** 出于历史兼容性考虑，该函数会有些返回结构在文档中并未提及（如 Config，Region 等），后续阿里云将逐步把这些字段删除，请您在调用 API 时不要依赖这些在文档中没有涉及的返回字段。

请求示例

```
https://r-kvstore.example.com
?<公共请求参数>
&Action=CreateInstance
&RegionId=cn-hangzhou
&Password=Pass!123456
&InstanceClass=redis.master.small.default
```

返回示例

XML格式：

```
<CreateInstanceResponse>
  <InstanceId>r-n8r8c5119d2e****</InstanceId>
  <InstanceName>testdb</InstanceName>
</CreateInstanceResponse>
```

JSON格式：

```
{
  "InstanceId": "r-n8r8c5119d2e****",
  "InstanceName": "testdb"
}
```

2.7.2. DeleteInstance

调用DeleteInstance接口释放Redis实例，调用时要求实例状态为运行中。

请求参数

请求参数

名称	类型	是否必须	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	DeleteInstance	系统规定参数，取值：DeleteInstance。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例 ID（全局唯一）。

返回参数

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见 公共返回参数 。

请求示例

```
https://r-kvstore.example.com
&Action=DeleteInstance
&InstanceId=r-n8r8c5119d2e****
&<公共请求参数>
```

返回示例

XML格式：

```
<DeleteInstanceResponse>
  <RequestId>5D622714-AEDD-4609-9167-F5DDD3D190D2</RequestId>
</DeleteInstanceResponse>
```

JSON格式：

```
{
  "RequestId": "5D622714-AEDD-4609-9167-F5DDD3D190D2"
}
```

2.7.3. ModifyInstanceSpec

调用ModifyInstanceSpec接口变更Redis实例的规格。

请求参数

请求参数

名称	类型	是否必须	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	ModifyInstanceSpec	操作接口名，取值：ModifyInstanceSpec。
InstanceId	String	是	r-bp1zxszhcgatnx****	实例 ID。
InstanceClass	String	是	redis.logic.sharding.drredisdb1g.1db.0rodb.4proxy.default	Redis实例的规格，更多信息，请参见 实例规格表 。

返回参数

返回参数

名称	类型	示例值	描述
<公共请求参数>	无	无	参见 公共返回参数 。
OrderId	String	20722623431****	订单ID。

请求示例

```
https://r-kvstore.example.com
&Action=ModifyInstanceSpec
&InstanceId=r-n8r8c5119d2e****
&InstanceClass= redis.master.mid.default
&<公共请求参数>
```

返回示例

XML格式：

```
<ModifyInstanceSpecResponse>
  <RequestId>A4828C10-AF0A-4CD7-8FBE-737C6CC1193E</RequestId>
  <OrderId>20722623431****</OrderId>
</ModifyInstanceSpecResponse>
```

JSON格式：

```
{
  "RequestId": "A4828C10-AF0A-4CD7-8FBE-737C6CC1193E",
  "OrderId": "20722623431****"
}
```

2.8. 实例管理

2.8.1. DescribeInstanceAttribute

调用DescribeInstanceAttribute查询Redis实例的详细信息。

请求参数

名称	类型	是否必选	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	DescribeInstanceAttribute	系统规定参数，取值：DescribeInstanceAttribute
InstanceId	String	是	r-bp1zxszhcgatnx**	实例ID。


返回参数

名称	类型	示例值	描述
Instances	List	无	由DBInstanceAttribute组成的数组。

DBInstanceAttribute 的参数构成

名称	类型	示例值	描述
InstanceId	String	r-bp1d72gwl41z7f****	实例ID。
InstanceName	String	apitest	实例名称。
Capacity	Long	1024	申请到的存储实例容量；单位：MB。
InstanceClass	String	redis.logic.sharding.drredisdb1g.1db.0rodb.4proxy.default	实例规格。
Bandwidth	Long	10	实例带宽限制；单位：Mbps。
Connections	Long	10000	实例连接数限制；单位：个。

名称	类型	示例值	描述
ConnectionDomain	String	r-bp1d72gwl41z7f****.redis.rds.aliyuncs.com	Redis 实例的连接域名（仅支持内网访问）。
Port	Int	6379	Redis连接端口。
RegionId	String	cn-hangzhou	申请的存储实例所在区域。参见Redis区域列表。
ZoneId	String	cn-hangzhou-b	RegionId下的可用区编码。参见Redis区域列表。
InstanceStatus	String	Normal	实例状态： <ul style="list-style-type: none"> Normal：正常 Creating：创建中 Changing：修改中 Inactive：被禁用
CreateTime	String	2019-03-06T10:42:03Z	实例创建时间。采用ISO8601表示法，并使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。
EndTime	String	2019-04-06T10:42:03Z	预付费实例到期时间。采用ISO8601表示法，并使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。
NetworkType	String	VPC	网络类型：CLASSIC或VPC。
VpcId	String	vpc-bp1nme44gek34slfc****	VPC ID。
VSwitchId	String	vsw-bp1e7clcw529l773d****	虚拟交换机ID。
MaintainStartTime	String	18:00Z	可运维开始时间。
MaintainEndTime	String	22:00Z	可运维结束时间。
SecurityIPList	String	127.0.0.1	IP白名单。
AvailabilityValue	String	100%	当月的可用性指标。

 **说明** 出于历史兼容性考虑，该函数会有些返回结构在文档中并未提及（如 Config、Region 等），后续阿里云将逐步把这些字段删除，请您在调用 API 时不要依赖这些在文档中没有涉及的返回字段。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=DescribeInstanceAttribute
&InstanceId=r-bplzxszhcgatnx****
&<公共请求参数>
```

返回示例

XML格式：

```
<DescribeInstanceAttributeResponse>
  <Instances>
    <DBInstanceAttribute>
      <Connections>10000</Connections>
      <ResourceGroupId>rg-acfmyiu4ekp****</ResourceGroupId>
      <Config>{"EvictionPolicy":"volatile-lru","hash-max-ziplist-entries":512,"zset-max-ziplist-entries":128,"list-max-ziplist-entries":512,"list-max-ziplist-value":64,"zset-max-ziplist-value":64,"set-max-intset-entries":512,"hash-max-ziplist-value":64}</Config>
    <
      <Port>6379</Port>
      <HasRenewChangeOrder>>false</HasRenewChangeOrder>
      <ZoneType>singlezone</ZoneType>
      <ConnectionDomain>r-bplwcw2rlw76acc5k****.redis.rds.aliyuncs.com</ConnectionDomain>
      <MaintainEndTime>22:00Z</MaintainEndTime>
      <Capacity>1024</Capacity>
      <QPS>100000</QPS>
      <PrivateIp>172.16.49.***</PrivateIp>
      <SecurityIPList>127.0.0.1</SecurityIPList>
      <InstanceStatus>Normal</InstanceStatus>
      <NetworkType>VPC</NetworkType>
      <Bandwidth>10</Bandwidth>
      <PackageType>standard</PackageType>
      <InstanceReleaseProtection>>false</InstanceReleaseProtection>
      <InstanceType>Redis</InstanceType>
      <AvailabilityValue>100.0%</AvailabilityValue>
      <Engine>Redis</Engine>
      <MaintainStartTime>18:00Z</MaintainStartTime>
      <Tags>
        <Tag>
          <Value>社区版</Value>
          <Key>类型</Key>
        </Tag>
      </Tags>
      <ArchitectureType>standard</ArchitectureType>
      <EngineVersion>5.0</EngineVersion>
      <ZoneId>cn-hangzhou-f</ZoneId>
      <InstanceId>r-bplwcw2rlw76ac****</InstanceId>
    </DBInstanceAttribute>
  </Instances>
</DescribeInstanceAttributeResponse>
```

```

        <VSwitchId>vsw-bplw9oue2nm66qlz****</VSwitchId>
        <CreateTime>2020-03-31T14:56:22Z</CreateTime>
        <InstanceClass>redis.logic.sharding.drredisdblg.1db.0rodb.4proxy.default
    </InstanceClass>
        <AuditLogRetention>7</AuditLogRetention>
        <IsRds>true</IsRds>
        <ReplicationMode>master-slave</ReplicationMode>
        <InstanceName>社区主从50</InstanceName>
        <VpcId>vpc-bp1opxulzkhn00gzv****</VpcId>
        <ChargeType>PostPaid</ChargeType>
        <VpcAuthMode>Open</VpcAuthMode>
        <NodeType>double</NodeType>
        <VpcCloudInstanceId>r-bplwcw2rlw76ac****</VpcCloudInstanceId>
        <RegionId>cn-hangzhou</RegionId>
    </DBInstanceAttribute>
</Instances>
    <RequestId>D67A69DF-C1FC-4B01-B239-B17888CCFD2A</RequestId>
</DescribeInstanceAttributeResponse>

```

JSON格式：

```

{
  "Instances": {
    "DBInstanceAttribute": [
      {
        "Connections": 10000,
        "ResourceGroupId": "rg-acfmyiu4ekp****",
        "Config": "{\"EvictionPolicy\":\"volatile-lru\",\"hash-max-ziplist-entries\":512,\"zset-max-ziplist-entries\":128,\"list-max-ziplist-entries\":512,\"list-max-ziplist-value\":64,\"zset-max-ziplist-value\":64,\"set-max-intset-entries\":512,\"hash-max-ziplist-value\":64}",
        "Port": 6379,
        "HasRenewChangeOrder": "false",
        "ZoneType": "singlezone",
        "ConnectionDomain": "r-bplwcw2rlw76acc5k****.redis.rds.aliyuncs.com",
        "MaintainEndTime": "22:00Z",
        "Capacity": 1024,
        "QPS": 100000,
        "PrivateIp": "172.16.49.***",
        "SecurityIPList": "127.0.0.1",
        "InstanceStatus": "Normal",
        "NetworkType": "VPC",
        "Bandwidth": 10,
        "PackageType": "standard",
        "InstanceReleaseProtection": false,
        "InstanceType": "Redis",
        "AvailabilityValue": "100.0%",
        "Engine": "Redis",
        "MaintainStartTime": "18:00Z",
        "Tags": {
          "Tag": [
            {
              "Value": "社区版",
              "Key": "类型"
            }
          ]
        }
      }
    ]
  }
}

```

```
    }
  ]
},
"ArchitectureType": "standard",
"EngineVersion": "5.0",
"ZoneId": "cn-hangzhou-f",
"InstanceId": "r-bplwcw2rlw76ac****",
"VSwitchId": "vsw-bplw9ouei2nm66qlz****",
"CreateTime": "2020-03-31T14:56:22Z",
"InstanceClass": "redis.logic.sharding.drredisdb1g.1db.0rodb.4proxy.default",
"AuditLogRetention": "7",
"IsRds": true,
"ReplicationMode": "master-slave",
"InstanceName": "社区主从50",
"VpcId": "vpc-bp1opxulzkh00gzv****",
"ChargeType": "PostPaid",
"VpcAuthMode": "Open",
"NodeType": "double",
"VpcCloudInstanceId": "r-bplwcw2rlw76ac****",
"RegionId": "cn-hangzhou"
}
]
},
"RequestId": "D67A69DF-C1FC-4B01-B239-B17888CCFD2A"
}
```

2.8.2. DescribeInstances

调用DescribeInstances接口查询账户下一个或多个实例的信息。

请求参数

请求参数

名称	类型	是否必须	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	DescribeInstances	操作接口名，取值：DescribeInstances。
InstanceIds	String	否	r-n8r8c5119d2e****	实例 ID（全局唯一）。 <div><div>说明</div><ul style="list-style-type: none">如需传入多个实例ID，使用英文逗号（,）分隔。不传入本参数即表示查询该用户名下所有实例。</div>

名称	类型	是否必须	示例值	描述
InstanceStatus	String	否	Normal	按状态过滤所需返回的实例： <ul style="list-style-type: none">• Normal：正常• Creating：创建中• Changing：修改中• Inactive：被禁用
RegionId	String	是	cn-hangzhou	调用 DescribeRegions 查询 RegionId。
InstanceType	String	否	Redis	引擎类型的过滤。 <ul style="list-style-type: none">• Memcache：Memcache 类型的实例• Redis：Redis 类型的实例
PageNumber	Integer	否	1	实例状态列表的页码，起始值为1，默认值为1。
PageSize	Integer	否	10	分页查询时设置的每页行数，最大值50行，默认值为10。
NetworkType	String	否	VPC	按网络类型过滤： <ul style="list-style-type: none">• CLASSIC：经典网络类型• VPC：VPC 网络类型

返回参数

返回参数


名称	类型	示例值	描述
<公共返回参数>	无	无	参见公共返回参数。
Instances	List	无	由 Instance 组成的数组，详见Instance参数构成。
TotalCount	Integer	1	实例总个数
PageNumber	Integer	1	实例列表的页码
PageSize	Integer	10	输入时设置的每页行数

Instance参数构成

名称	类型	示例值	描述
InstanceId	String	r-n8r8c5119d2e** **	实例 ID（全局唯一）

名称	类型	示例值	描述
InstanceName	String	testdb	实例名称
Capacity	Long	512	申请到的存储实例容量；单位：MB。
InstanceClass	String	redis.master.mid.default	实例规格信息
Bandwidth	Long	128	实例带宽限制；单位：Mbps。
Connections	Long	300	实例连接数限制；单位：个。
ConnectionDomain	String	de5d88e34d00***.m.cn-hzalicm10pub001.r-kvstore.example.com	实例连接域名（仅支持内网访问。）
Port	Int	11211	连接端口
UserName	String	testuser	连接用户名
RegionId	String	cn-qingdao	申请的存储实例所在区域
ZoneId	String	cn-qingdao-b	RegionId 下的可用区编码
InstanceStatus	String	Normal	实例状态： <ul style="list-style-type: none"> Normal：正常 Creating：创建中 Changing：修改中 Inactive：被禁用 Transforming：转换中 BackupRecovering：备份恢复中 MinorVersionUpgrading：实例小版本升级中
ChargeType	String	PostPaid	付费类型：PrePaid 或 PostPaid
CreateTime	String	2019-03-06T10:42:03Z	实例创建时间。采用ISO8601表示法，并使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。
EndTime	String	2019-04-06T10:42:03Z	预付费实例到期时间。采用ISO8601表示法，并使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。
NetworkType	String	VPC	网络类型：CLASSIC 或 VPC

名称	类型	示例值	描述
VpcId	String	vpc-bp1nme44gek34slfc****	VPC 的 ID
VSwitchId	String	vsw-bp1e7clcw529l773d****	虚拟交换机 ID
PrivateIpAddr	String	172.16.**.*	私有 IP 地址

 **说明** 出于历史兼容性考虑，该函数会有些返回结构在文档中并未提及（如 Config），后续阿里云将逐步把这些字段删除，请您在调用 API 时不要依赖这些在文档中没有涉及的返回字段。

请求示例

```
https://r-kvstore.example.com
?<公共请求参数>
&Action=DescribeInstances
&PageNumber=1
&PageSize=10
&InstanceId=r-n8r8c5119d2e****
```

返回示例

XML格式：

```
<DescribeInstancesResponse>
  <Instances>
    <Instance>
      <Bandwidth>128</Bandwidth>
      <Capacity>512</Capacity>
      <ConnectionDomain>de5d88e34d00****.m.cnhzalicm10pub001.r-kvstore.example.com</C
onnectionDomain>
      <Connections>300</Connections>
      <ZoneId>cn-qingdao-b</ZoneId>
      <InstanceId>r-n8r8c5119d2e****</InstanceId>
      <InstanceName>testdb</InstanceName>
      <InstanceStatus>Available</InstanceStatus>
      <InstanceClass>redis.master.mid.default</InstanceClass>
      <Port>11211</Port>
      <RegionId>cn-qingdao</RegionId>
      <UserName>testuser</UserName>
      <NetworkType>CLASSSSIC</NetworkType>
      <ChargeType> PostPaid</ChargeType>
    </Instance>
  </Instances>
  <PageNumber>1</PageNumber>
  <PageSize>10</PageSize>
  <TotalCount>1</TotalCount>
  <RequestId>969D0A1D-C91A-4837-9F70-49785DF9BDCE</RequestId>
</DescribeInstancesResponse>
```

JSON格式：

```
{
  "Instances" : {
    "Instance" : [{
      "Bandwidth" : 128,
      "Capacity" : 512,
      "ConnectionDomain" : "de5d88e34d00****.m.cnhzalicm10pub001.r-kvstore.examp
e.com",
      "Connections" : 300,
      "ZoneId" : "cn-qingdao-b",
      "InstanceId" : "r-n8r8c5119d2e****",
      "InstanceName" : "testdb",
      "InstanceStatus" : "Available",
      "InstanceClass" : "redis.master.mid.default",
      "Port" : 11211,
      "RegionId" : "cn-qingdao",
      "UserName" : "testuser"
      "NetworkType": "CLASSIC"
    "ChargeType": "PostPaid"
    }
  ],
  "PageNumber" : 1,
  "PageSize" : 10,
  "TotalCount" : 1,
  "RequestId" : "969D0A1D-C91A-4837-9F70-49785DF9BDCE"
}
```

2.8.3. DescribeRegions

调用DescribeRegions查询可创建Redis实例的地域。

请求参数

名称	类型	是否必须	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	DescribeRegions	系统规定参数，取值：DescribeRegions。

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见 公共返回参数 。

名称	类型	示例值	描述
RegionIds	List	无	RegionIds 为一个 List，里面每个元素由两部分组成：RegionId、ZoneIds，其中的 ZoneIds 也是一个 String，多个 ZoneId 用 “,” 分隔。

请求示例

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

返回示例

XML格式：

```
<DescribeRegionsResponse>
  <RegionIds>
    <KVStoreRegion>
      <RegionId>cn-hangzhou</RegionId>
      <ZoneIdList>
        <ZoneId>cn-hangzhou-h</ZoneId>
        <ZoneId>cn-hangzhou-MAZ10(h,i)</ZoneId>
        <ZoneId>cn-hangzhou-MAZ9(g,h)</ZoneId>
        <ZoneId>cn-hangzhou-b</ZoneId>
        <ZoneId>cn-hangzhou-d</ZoneId>
        <ZoneId>cn-hangzhou-e</ZoneId>
        <ZoneId>cn-hangzhou-MAZ6(b,f)</ZoneId>
        <ZoneId>cn-hangzhou-f</ZoneId>
        <ZoneId>cn-hangzhou-g</ZoneId>
      </ZoneIdList>
      <RegionEndpoint>r-kvstore.aliyuncs.com</RegionEndpoint>
      <ZoneIds>cn-hangzhou-h,cn-hangzhou-MAZ10(h,i),cn-hangzhou-MAZ9(g,h),cn-hangzhou-b,cn-hangzhou-d,cn-hangzhou-e,cn-hangzhou-MAZ6(b,f),cn-hangzhou-f,cn-hangzhou-g</ZoneIds>
    </KVStoreRegion>
  </RegionIds>
  <LocalName>华东1（杭州）</LocalName>
  <RequestId>AD425AD3-CC7B-4EE2-A5CB-2F61BA73****</RequestId>
</DescribeRegionsResponse>
```

JSON格式：

```
{
  "RegionIds": {
    "KVStoreRegion": [{
      "RegionId": "cn-hangzhou",
      "ZoneIdList": {
        "ZoneId": [
          "cn-hangzhou-h",
          "cn-hangzhou-MAZ10(h,i)",
          "cn-hangzhou-MAZ9(g,h)",
          "cn-hangzhou-b",
          "cn-hangzhou-d",
          "cn-hangzhou-e",
          "cn-hangzhou-MAZ6(b,f)",
          "cn-hangzhou-f",
          "cn-hangzhou-g"
        ]
      },
      "RegionEndpoint": "r-kvstore.aliyuncs.com",
      "ZoneIds": "cn-hangzhou-h,cn-hangzhou-MAZ10(h,i),cn-hangzhou-MAZ9(g,h),cn-hangzhou-b,cn-hangzhou-d,cn-hangzhou-e,cn-hangzhou-MAZ6(b,f),cn-hangzhou-f,cn-hangzhou-g",
      "LocalName": "华东1（杭州）"
    }
  ],
  "RequestId": "AD425AD3-CC7B-4EE2-A5CB-2F61BA73****"
}
```

2.8.4. DescribeSecurityIps

调用DescribeSecurityIps接口查询Redis实例的IP白名单。

请求参数

名称	类型	是否必选	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	DescribeSecurityIps	系统规定参数，取值：DescribeSecurityIps。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例ID。

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见 公共返回参数 。
SecurityIpGroups	List	无	实例的IP白名单分组列表。

SecurityIpGroups 的参数

名称	类型	示例值	描述
SecurityGroupName	String	default	IP 白名单分组的名字。
SecurityIpList	String	100.100.202.*/24,10.10.10.*/24	IP 白名单分组下的 IP 列表，最多1000个。IP 之间以逗号隔开，格式如下： 0.0.0.0/0,10.23.12.24 (IP) 或者 10.23.12.24/24 (CIDR 模式，无类域间路由，/24表示了地址中前缀的长度，范围[1,32])。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=DescribeSecurityIps
&InstanceId=r-bplzxszhcgatnx****
&<公共请求参数>
```

返回示例

XML格式：

```
<DescribeSecurityIpsResponse>
  <SecurityIpGroups>
    <SecurityIpGroup>
      <SecurityIpList>127.0.0.1</SecurityIpList>
      <SecurityIpGroupAttribute></SecurityIpGroupAttribute>
      <SecurityGroupName>default</SecurityGroupName>
    </SecurityIpGroup>
    <SecurityIpGroup>
      <SecurityIpList>100.100.202.*/24,10.10.10.*/24</SecurityIpList>
      <SecurityIpGroupAttribute>hidden</SecurityIpGroupAttribute>
      <SecurityGroupName>rds_replica_group</SecurityGroupName>
    </SecurityIpGroup>
  </SecurityIpGroups>
  <RequestId>EFC9161F-15E3-4A6E-8A99-C09916D1****</RequestId>
</DescribeSecurityIpsResponse>
```

JSON格式：

```
{
  "SecurityIpGroups": {
    "SecurityIpGroup": [
      {
        "SecurityIpList": "127.0.0.1",
        "SecurityIpGroupAttribute": "",
        "SecurityIpGroupName": "default"
      },
      {
        "SecurityIpList": "100.100.202.***/24,10.10.10.***",
        "SecurityIpGroupAttribute": "hidden",
        "SecurityIpGroupName": "rds_replica_group"
      }
    ]
  },
  "RequestId": "EFC9161F-15E3-4A6E-8A99-C09916D1****"
}
```

2.8.5. FlushInstance

调用FlushInstance清空Redis实例中的数据，不可恢复。

请求参数

请求参数

名称	类型	是否必选	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	FlushInstance	系统规定参数，取值：FlushInstance。
InstanceId	String	是	r- bp1zxszhcgatnx**	实例 ID，全局唯一。

返回参数

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见 公共返回参数 。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=FlushInstance
&InstanceId=r-bp1zxszhcgatnx****
&<公共请求参数>
```

返回示例

XML格式：

```
<FlushInstanceResponse>
  <RequestId>8D0C0AFC-E9CD-47A4-8395-5C31BF9B3E76</RequestId>
</FlushInstanceResponse>
```

JSON格式：

```
{
  "RequestId": "8D0C0AFC-E9CD-47A4-8395-5C31BF9B3E7"
}
```

2.8.6. ModifyInstanceAttribute

修改实例的属性，包括名称或密码。

请求参数

请求参数

名称	类型	是否必须	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	ModifyInstanceAttribute	操作接口名，取值：ModifyInstanceAttribute。
InstanceId	String	是	r-bp1zxszhcgatnx****	实例 ID，全局唯一。
InstanceName	String	否	TestInstance	修改后的实例名称。名称为2-128个字符，以大小写英文字母或中文开头，不支持字符@/:=" <>{}和空格。
NewPassword	String	否	Qa12345678	实例密码。字符类型：8~30个字符，需同时包含大写字母、小写字母和数字。 <div> 说明 对于特殊字符 !<>()[];`~.-_@#%\$^&*暂不支持。</div>

返回参数

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见公共返回参数。

请求示例

```
https://r-kvstore.example.com
&Action=ModifyInstanceAttribute
&InstanceId=r-bplzxszhcgatnx****
&NewPassword=Qa12345678
&InstanceName=TestInstance
&<公共请求参数>
```

返回示例

XML格式：

```
<ModifyInstanceAttributeResponse>
  <RequestId>E3B35BEA-9EB0-402C-88CF-C46CCCC1EE59</RequestId>
</ModifyInstanceAttributeResponse>
```

JSON格式：

```
{
  "RequestId" : "E3B35BEA-9EB0-402C-88CF-C46CCCC1EE59"
}
```

2.8.7. ModifyInstanceMaintainTime

调用ModifyInstanceMaintainTime修改Redis实例的可维护时间段，阿里云将在您设定的可维护时间段内对Redis实例进行例行维护。

请求参数

请求参数

名称	类型	是否必须	示例值	描述
<公共请求参数>	无	是	无	参见公共请求参数。
Action	String	是	ModifyInstanceMaintainTime	系统规定参数，取值：ModifyInstanceMaintainTime。

名称	类型	是否必须	示例值	描述
InstanceId	String	是	r-bp1xszhcgatnx**	实例ID。
MaintainStartTime	String	是	03:00Z	可运维时间开始时间。
MaintainEndTime	String	是	04:00Z	可运维时间结束时间。

返回参数

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见 公共返回参数 。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=ModifyInstanceMaintainTime
&InstanceId=r-bp1xszhcgatnx****
&MaintainStartTime=17:00Z
&MaintainEndTime=18:00Z
&<公共请求参数>
```

返回示例

XML格式：

```
<ModifyInstanceMaintainTimeResponse>
  <RequestId>8D0C0AFC-E9CD-47A4-8395-5C31BF9B3E76</RequestId>
</ModifyInstanceMaintainTimeResponse>
```

JSON格式：

```
{
  "RequestId": "8D0C0AFC-E9CD-47A4-8395-5C31BF9B3E76"
}
```

2.8.8. ModifyInstanceNetExpireTime

若Redis实例之前执行过由经典网络向VPC网络切换，并保留了经典网络连接地址，则可调用ModifyInstanceNetExpireTime延长经典网络连接地址的保留时间。

请求参数

名称	类型	是否必选	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	ModifyInstanceNetExpireTime	系统规定参数，取值：ModifyInstanceNetExpireTime。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例ID。
ConnectionString	String	是	r-bp1zxszhcgatnx** *.redis.rds.aliyuncs.com	经典网络的访问域名。
ClassicExpiredDays	String	是	14	选择保留时长。取值：14、30、60或120。

返回参数

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见 公共返回参数 。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=ModifyInstanceNetExpireTime
&ClassicExpiredDays=14
&ConnectionString=r-bp1zxszhcgatnx***.redis.rds.aliyuncs.com
&InstanceId=r-bp1zxszhcgatnx***
&<公共请求参数>
```

返回示例

XML格式：

```
<ModifyInstanceNetExpireTimeResponse>
  <RequestId>9C4AF387-1EA3-4C84-8013-3F6B973EDDF5</RequestId>
</ModifyInstanceNetExpireTimeResponse>
```

JSON格式：

```
{
  "RequestId": "9C4AF387-1EA3-4C84-8013-3F6B973EDDF5",
}
```


2.8.9. ModifySecurityIps

调用ModifySecurityIps设置Redis实例的IP白名单。

请求参数

请求参数

名称	类型	是否必选	示例值	描述
<公共请求参数>	无	是	无	参见公共请求参数 公共参数 。
Action	String	是	ModifySecurityIps	系统规定参数，取值：ModifySecurityIps。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例ID（全局唯一）。
SecurityIps	String	是	100.64.***.0/24,10.101.11.***	待修改的IP白名单列表。
SecurityIpsGroupName	String	是	default	白名单分组。

返回参数

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见 公共返回参数 。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=ModifySecurityIps
&InstanceId=r-bp1zxszhcgatnx****
&SecurityIps=100.64.***.0/24,10.101.11.***
&<公共请求参数>
```

返回示例

XML格式：

```
<ModifySecurityIpsResponse>
  <RequestId>AAAF99B1-69ED-4E80-8CD5-272C09E4****</RequestId>
</ModifySecurityIpsResponse>
```

JSON格式：

```
{
  "RequestId" : "AAAF99B1-69ED-4E80-8CD5-272C09E4****"
}
```

2.8.10. SwitchNetwork

调用SwitchNetwork切换Redis实例的专有网络VPC或交换机，如果Redis实例为经典网络，则会将其切换为专有网络。

请求参数

名称	类型	是否必须	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	SwitchNetwork	系统规定参数，取值：SwitchNetwork。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例ID。
TargetNetworkType	String	是	VPC	切换到的网络类型。 <ul style="list-style-type: none">VPC：专有网络类型的实例Classic：经典网络类型的实例 当前只支持经典网络切换到 VPC 类型，该类型需要填写 VPC。
VpcId	String	否	vpc-bp1nme44gek34slfc****	切换到的 VPC 的虚拟网络ID。
VSwitchId	String	否	vsw-bp1e7clcw529l773d****	切换到的 VPC 的虚拟交换机 ID。如果指定了 VPC ID，则该参数也必须指定。
RetainClassic	String	否	True	是否保留经典网络 IP。默认是False。 <ul style="list-style-type: none">True：保留False：不保留

名称	类型	是否必须	示例值	描述
ClassicExpiredDays	String	否	30	保留经典网络 IP 的时间。单位：天。 可选值：14、30、60、120。若选择保留经典网络地址，则该参数必选。

返回参数

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见公共返回参数。

请求示例

```
https://r-kvstore.aliyuncs.com?Action=SwitchNetwork
&InstanceId=r-bplzxszhcgatnx****
&TargetNetworkType=VPC
&VpcId=vpc-bplnme44gek34slfc****
&VSwitchId=vsw-bple7clcw529l773d****
&RetainClassic=True
&ClassicExpiredDays=30
&<公共请求参数>
```

返回示例

XML格式：

```
<SwitchNetworkResponse>
  <TaskId>578678678</TaskId>
  <requestId>F0997EE8-F4C2-4503-9168-85177ED78C70</requestId>
</SwitchNetworkResponse>
```

JSON格式：

```
{
  "TaskId": "578678678",
  "requestId": "F0997EE8-F4C2-4503-9168-85177ED78C70"
}
```

2.9. 备份恢复

2.9.1. CreateBackup

调用CreateBackup为Redis实例创建数据备份。

请求参数

名称	类型	是否必选	示例值	描述
<公共请求参数>	无	是	无	参见公共请求参数。
Action	String	是	CreateBackup	系统规定参数，取值：CreateBackup。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例ID。

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见公共返回参数。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=CreateBackup
&InstanceId=r-bp1zxszhcgatnx****
&<公共请求参数>
```

返回示例

XML格式：

```
<CreateBackupResponse>
  <RequestId>2FF6158E-3394-4A90-B634-79C49184****</RequestId>
  <BackupJobID>1162****</BackupJobID>
</CreateBackupResponse>
```

JSON格式：

```
{
  "RequestId": "2FF6158E-3394-4A90-B634-79C49184****",
  "BackupJobID": "1162****"
}
```

2.9.2. ModifyBackupPolicy

调用ModifyBackupPolicy修改Redis实例的自动备份策略。

请求参数

名称	类型	是否必选	示例值	描述
<公共请求参数>	无	是	无	参见公共请求参数。
Action	String	是	ModifyBackupPolicy	系统规定参数，取值：ModifyBackupPolicy。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例ID。
PreferredBackupTime	String	是	00:00Z-04:00Z	备份时间，格式：HH:mmZ-HH:mm Z。
PreferredBackupPeriod	String	是	Tuesday	备份周期： <ul style="list-style-type: none">Monday: 周一Tuesday: 周二Wednesday: 周三Thursday: 周四Friday: 周五Saturday: 周六Sunday: 周日

返回参数

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见公共返回参数。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=ModifyBackupPolicy
&InstanceId=r-bp1zxszhcgatnx****
&PreferredBackupTime=00:00Z-04:00Z
&PreferredBackupPeriod=Tuesday
&<公共请求参数>
```

返回示例

XML格式：

```
<ModifyBackupPolicyResponse>
  <RequestId>8D0C0AFC-E9CD-47A4-8395-5C31BF9B****</RequestId>
</ModifyBackupPolicyResponse>
```

JSON格式：

```
{
  "RequestId": "8D0C0AFC-E9CD-47A4-8395-5C31BF9B****"
}
```

2.9.3. DescribeBackupPolicy

调用DescribeBackupPolicy查询Redis实例的备份策略，包括备份周期、备份时间等。

请求参数

名称	类型	是否必选	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	DescribeBackupPolicy	系统规定参数，取值：DescribeBackupPolicy。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例ID。

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见 公共返回参数 。
BackupRetentionPeriod	String	7	备份保留天数。
PreferredBackupTime	String	05:00Z-06:00Z	备份时间，格式：HH:mmZ- HH:mm Z。
PreferredBackupPeriod	String	Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday	备份周期： <ul style="list-style-type: none">Monday: 周一Tuesday: 周二Wednesday: 周三Thursday: 周四Friday: 周五Saturday: 周六Sunday: 周日

名称	类型	示例值	描述
PreferredNextBackupTime	String	2019-03-14T05:28Z	下次备份时间

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=DescribeBackupPolicy
&InstanceId=r-bplzxszhcgatnx****
&<公共请求参数>
```

返回示例

XML格式：

```
<DescribeBackupPolicyResponse>
  <PreferredBackupPeriod>Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday</PreferredBackupPeriod>
  <RequestId>90B82DB7-FB28-4CC2-ADBF-1F8659F3****</RequestId>
  <PreferredNextBackupTime>2018-10-26T19:05Z</PreferredNextBackupTime>
  <PreferredBackupTime>19:00Z-20:00Z</PreferredBackupTime>
  <BackupRetentionPeriod>7</BackupRetentionPeriod>
</DescribeBackupPolicyResponse>
```

JSON格式：

```
{
  "PreferredBackupPeriod": "Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday",
  "RequestId": "90B82DB7-FB28-4CC2-ADBF-1F8659F3****",
  "PreferredNextBackupTime": "2018-10-26T19:05Z",
  "PreferredBackupTime": "19:00Z-20:00Z",
  "BackupRetentionPeriod": "7"
}
```

2.9.4. DescribeBackups

调用DescribeBackups查询Redis实例的备份文件信息。

请求参数

名称	类型	是否必选	示例值	描述
<公共请求参数>	无	是	无	参见公共请求参数。
Action	String	是	DescribeBackups	系统规定参数，取值：DescribeBackups。

名称	类型	是否必选	示例值	描述
InstanceId	String	是	r-bp1xszhcgatnx**	实例ID。
BackupId	String	否	11611111	备份集ID。
PageSize	String	是	30	每页记录数，取值：30/50/100。
PageNumber	String	是	1	页码，大于0，且不超过 Integer 的最大值。
StartTime	String	是	2019-03-11T10:00Z	查询开始时间。
EndTime	String	是	2019-03-14T18:00Z	查询结束时间。

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见公共返回参数。
Backups	List	无	由Backup组成的数组。
PageNumber	Integer	1	页码。
TotalCount	Integer	5	备份总个数。
PageSize	Integer	30	每页记录数。

Backup 的参数构成

名称	类型	示例值	描述
BackupId	String	11111111	备份集ID。
BackupDBNames	String	all	备份的DB名称。
BackupStatus	String	Success	备份集状态： <ul style="list-style-type: none">Success：成功Failed：失败
BackupStartTime	String	2019-03-14T05:28:50Z	备份开始时间。
BackupEndTime	String	2019-03-14T05:31:13Z	备份结束时间。

名称	类型	示例值	描述
BackupType	String	FullBackup	备份类型： <ul style="list-style-type: none">FullBackup：全量备份IncrementalBackup：增量备份
BackupMode	String	Automated	备份模式： <ul style="list-style-type: none">Automated：自动备份Manual：手工触发备份
BackupMethod	String	Physical	备份方法： <ul style="list-style-type: none">Logical：逻辑备份Physical：物理备份
BackupDownloadURL	String	https://rdsbak-hk45-v2.oss-cn-hongkong.aliyuncs.com/*****	备份文件的URL下载地址。
BackupSize	String	1024	备份集大小。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=DescribeBackups
&InstanceId=r-bplzxszhcgatnx****
&StartTime=2019-03-11T10:00Z
&EndTime=2019-03-14T18:00Z
&<公共请求参数>
```

返回示例

XML格式：

```
<DescribeBackupsResponse>
  <TotalCount>1</TotalCount>
  <RequestId>721E2D39-351D-40E2-BF7C-B5075B0A****</RequestId>
  <PageSize>30</PageSize>
  <PageNumber>1</PageNumber>
  <Backups>
    <Backup>
      <BackupMethod>Physical</BackupMethod>
      <EngineVersion>4.0</EngineVersion>
      <BackupIntranetDownloadURL>http://rdsbak-hz-v3.oss-cn-hangzhou-internal.aliyuncs.com/*****</BackupIntranetDownloadURL>
      <BackupMode>Automated</BackupMode>
      <BackupSize>2263040</BackupSize>
      <BackupId>775661111</BackupId>
      <NodeInstanceId>r-bp12fdcf4dea****</NodeInstanceId>
      <BackupDBNames>all</BackupDBNames>
      <BackupDownloadURL>http://rdsbak-hz-v3.oss-cn-hangzhou.aliyuncs.com/*****</BackupDownloadURL>
      <BackupEndTime>2020-10-30T17:31:26Z</BackupEndTime>
      <BackupStartTime>2020-10-30T17:28:28Z</BackupStartTime>
      <BackupType>FullBackup</BackupType>
      <BackupStatus>Success</BackupStatus>
    </Backup>
  </Backups>
</DescribeBackupsResponse>
```

JSON格式：

```
{
  "TotalCount": 1,
  "RequestId": "721E2D39-351D-40E2-BF7C-B5075B0A****",
  "PageSize": 30,
  "PageNumber": 1,
  "Backups": {
    "Backup": [
      {
        "BackupMethod": "Physical",
        "EngineVersion": "4.0",
        "BackupIntranetDownloadURL": "http://rdsbak-hz-v3.oss-cn-hangzhou-internal.aliyuncs.com/****",
        "BackupMode": "Automated",
        "BackupSize": 2263040,
        "BackupId": 775661111,
        "NodeInstanceId": "r-bp12fdcf4dea****",
        "BackupDBNames": "all",
        "BackupDownloadURL": "http://rdsbak-hz-v3.oss-cn-hangzhou.aliyuncs.com/****",
        "BackupEndTime": "2020-10-30T17:31:26Z",
        "BackupStartTime": "2020-10-30T17:28:28Z",
        "BackupType": "FullBackup",
        "BackupStatus": "Success"
      }
    ]
  }
}
```

2.9.5. RestoreInstance

调用RestoreInstance将备份文件中的数据恢复到当前Redis实例中，结合数据闪回更可实现将指定的Key恢复至某个秒级时间点。

请求参数

名称	类型	是否必须	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	RestoreInstance	系统规定参数，取值：RestoreInstance。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例ID。
BackupId	String	是	78241****	备份结果集ID。

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见公共返回参数。

请求示例

```
http(s)://r-kvstore.aliyuncs.com/?Action=RestoreInstance
&InstanceId=r-bplzxszhcgatnx****
&RestoreType=1
&RestoreTime=2021-07-06T07:25:57Z
&FilterKey=key:00000007198*
&<公共请求参数>
```

返回示例

XML格式：

```
<RestoreInstanceResponse>
  <RequestId>8D0C0AFC-E9CD-47A4-8395-5C31BF9B****</RequestId>
</RestoreInstanceResponse>
```

JSON格式：

```
{
  "RequestId": "8D0C0AFC-E9CD-47A4-8395-5C31BF9B****"
}
```

2.10. 监控管理

2.10.1. DescribeMonitorItems

调用DescribeMonitorItems查询Redis实例支持的监控项列表。

请求参数

请求参数

名称	类型	是否必须	示例值	描述
<公共请求参数>	无	是	无	参见公共请求参数。
Action	String	是	DescribeMonitorItems	系统规定参数，取值：DescribeMonitorItems。

返回参数

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见 公共返回参数 。
MonitorItems	List	无	每个可查看的监控参数List。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=DescribeMonitorItems
&<公共请求参数>
```

返回示例

XML格式：

```
<DescribeMonitorItemsResponse>
  <MonitorItems>
    <MonitorItem>
      <MonitorKey>GetQ</MonitorKey>
      <Unit>Counts/s</Unit>
    </MonitorItem>
    <MonitorItem>
      <MonitorKey>Flush</MonitorKey>
      <Unit>Counts/s</Unit>
    </MonitorItem>
    <MonitorItem>
      <MonitorKey>UsedMemCache</MonitorKey>
      <Unit>Bytes</Unit>
    </MonitorItem>
    <MonitorItem>
      <MonitorKey>ReplaceQ</MonitorKey>
      <Unit>Counts/s</Unit>
    </MonitorItem>
  </MonitorItems>
  <RequestId>B906A893-58A3-4644-AC2D-A2C9B08706C1</RequestId>
</DescribeMonitorItemsResponse>
```

JSON格式：

```
{
  "MonitorItems" : {
    "MonitorItem" : [{
      "MonitorKey" : "GetQ",
      "Unit" : "Counts/s"
    }, {
      "MonitorKey" : "Flush",
      "Unit" : "Counts/s"
    }, {
      "MonitorKey" : "UsedMemCache",
      "Unit" : "Bytes"
    }, {
      "MonitorKey" : "ReplaceQ",
      "Unit" : "Counts/s"
    }
  ]
},
"RequestId" : "B906A893-58A3-4644-AC2D-A2C9B08706C1"
}
```

2.10.2. DescribeHistoryMonitorValues

调用DescribeHistoryMonitorValues查看Redis实例的性能监控信息。

请求参数

名称	类型	是否必须	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	DescribeHistoryMonitorValues	系统规定参数，取值：DescribeHistoryMonitorValues。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例ID。
StartTime	String	是	2018-10-18T12:00:00Z	历史监控开始时间点；采用ISO8601表示法，并使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。

名称	类型	是否必须	示例值	描述
EndTime	String	是	2018-10-19T12:00:00Z	历史监控结束时间点；采用ISO8601表示法，并使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。EndTime 必须大于等于 StartTime。
IntervalForHistory	String	是	01m	取值必须是01m、05m、15m、60m。监控信息采样间隔，即每隔 N 分钟取一个监控采样点。
MonitorKeys	String	否	CpuUsage	监控项Key。通过函数 DescribeMonitorItems 查询。

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见公共返回参数。
MonitorHistory	String	<pre>{ "2020-11-18T00:00:00Z": { "CpuUsage": "0.30" }, "2020-11-18T00:02:00Z": { "CpuUsage": "0.60" }, "2020-11-18T00:03:00Z": { "CpuUsage": "0.60" } }</pre>	以JSON格式返回的监控信息。返回的监控项，参见 DescribeMonitorItems。 <div>说明 为了提高数据传输效率，只有非0的监控数据才会返回，其余未显示给出的监控数据均为默认值0。</div>

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=DescribeHistoryMonitorValues
&InstanceId=r-bplzxszhcgatnx****
&StartTime=2018-10-18T12:00:00Z
&EndTime=2018-10-18T12:03:00Z
&IntervalForHistory=01m
&MonitorKeys=CpuUsage
&<公共请求参数>
```

返回示例

XML格式：

```
<DescribeHistoryMonitorValuesResponse>
  <MonitorHistory>{"2020-11-18T00:00:00Z":{"CpuUsage":"0.30"},"2020-11-18T00:02:00Z":{"CpuUsage":"0.60"},"2020-11-18T00:03:00Z":{"CpuUsage":"0.60"}}</MonitorHistory>
  <RequestId>689E8093-3094-4DC2-A2A2-30D40097****</RequestId>
</DescribeHistoryMonitorValuesResponse>
```

JSON格式：

```
{
  "MonitorHistory": "{\"2020-11-18T00:00:00Z\":{\"CpuUsage\":\"0.30\"},\"2020-11-18T00:02:00Z\":{\"CpuUsage\":\"0.60\"},\"2020-11-18T00:03:00Z\":{\"CpuUsage\":\"0.60\"}}",
  "RequestId": "689E8093-3094-4DC2-A2A2-30D40097****"
}
```

2.11. 参数管理

2.11.1. DescribeInstanceConfig

调用DescribeInstanceConfig查询Redis实例的参数配置信息。

请求参数

请求参数

名称	类型	是否必选	示例值	描述
<公共请求参数>	无	是	无	参见公共请求参数。
Action	String	是	DescribeInstanceConfig	系统规定参数，取值：DescribeInstanceConfig。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例ID。

返回参数

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见公共返回参数。

名称	类型	示例值	描述
Config	String	<code>{"#no_loose_statistics-cmds":"SET,GET","EvictionPolicy":"volatile-lru","hash-max-ziplist-entries":512,"slowlog-log-slower-than":"1000","notify-keyspace-events":"\$","zset-max-ziplist-entries":128,"list-max-ziplist-entries":512,"list-max-ziplist-value":64,"zset-max-ziplist-value":64,"set-max-intset-entries":512,"hash-max-ziplist-value":64,"#no_loose_disabled-commands":"flushall"}</code>	实例的配置参数。参见 实例配置参数表 。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=DescribeInstanceConfig
&InstanceId=r-bplzxszhcgatnx****
&<公共请求参数>
```

返回示例

XML格式：

```
<DescribeInstanceConfigResponse>
  <RequestId>4E2C08F6-2D11-4ECD-9A4C-27EF2D3D****</RequestId>
  <Config>{"#no_loose_statistics-cmds":"SET,GET","EvictionPolicy":"volatile-lru","hash-max-ziplist-entries":512,"slowlog-log-slower-than":"1000","notify-keyspace-events":"$","zset-max-ziplist-entries":128,"list-max-ziplist-entries":512,"list-max-ziplist-value":64,"zset-max-ziplist-value":64,"set-max-intset-entries":512,"hash-max-ziplist-value":64,"#no_loose_disabled-commands":"flushall"}</Config>
</DescribeInstanceConfigResponse>
```

JSON格式：

```
{
  "RequestId": "4E2C08F6-2D11-4ECD-9A4C-27EF2D3D****",
  "Config": "{\"#no_loose_statistics-cmds\":\"SET,GET\", \"EvictionPolicy\":\"volatile-lru\", \"hash-max-ziplist-entries\":512, \"slowlog-log-slower-than\":\"1000\", \"notify-keyspace-events\":\"$\", \"zset-max-ziplist-entries\":128, \"list-max-ziplist-entries\":512, \"list-max-ziplist-value\":64, \"zset-max-ziplist-value\":64, \"set-max-intset-entries\":512, \"hash-max-ziplist-value\":64, \"#no_loose_disabled-commands\":\"flushall\"}"
}
```

2.11.2. ModifyInstanceConfig

调用ModifyInstanceConfig修改Redis实例的参数配置。

请求参数

请求参数

名称	类型	是否必选	示例值	描述
<公共请求参数>	无	是	无	参见 公共请求参数 。
Action	String	是	ModifyInstanceConfig	系统规定参数，取值：ModifyInstanceConfig。
InstanceId	String	是	r-bp1zxszhcgatnx**	实例 ID
Config	String	是	{"EvictionPolicy":"volatile-lru","list-max-ziplist-entries":512,"zset-max-ziplist-entries":128,"hash-max-ziplist-entries":512,"hash-max-ziplist-value":64,"list-max-ziplist-value":64,"set-max-intset-entries":512,"zset-max-ziplist-value":64}	实例的配置参数（Json String）。参见 实例配置参数表 。

返回参数

返回参数

名称	类型	示例值	描述
<公共返回参数>	无	无	参见公共返回参数。

请求示例

```
https://r-kvstore.aliyuncs.com/?Action=ModifyInstanceConfig
&InstanceId=r-bplzxszhcgatnx****
&Config={"EvictionPolicy":"volatile-lru","list-max-ziplist-entries":512,"zset-max-ziplist-e
ntries":128,"hash-max-ziplist-entries":512,"hash-max-ziplist-value":64,"list-max-ziplist-va
lue":64,"set-max-intset-entries":512,"zset-max-ziplist-value":64}
&<公共请求参数>
```

返回示例

XML格式：

```
<ModifyInstanceConfigResponse>
  <RequestId>8D0C0AFC-E9CD-47A4-8395-5C31BF9B****</RequestId>
</ModifyInstanceConfigResponse>
```

JSON格式：

```
{
  "requestId":"8D0C0AFC-E9CD-47A4-8395-5C31BF9B****"
}
```

2.12. 附表

2.12.1. 实例规格表

云数据库Redis支持社区版和企业版（性能增强型），社区版实例支持标准、集群和读写分离架构，企业版实例支持标准和集群架构。本文介绍规格的详细信息（包含内存容量、连接数、带宽等）。

带宽说明

- 在网络带宽资源充足的情况下，云数据库Redis版没有带宽限制；当资源不足时，实例的带宽限制才会生效。
- 集群实例的总带宽上限为2,048 MB/s。达到该上限后，即使选择拥有更多分片的规格，带宽也不会提高。
- 带宽分别应用于上行带宽和下行带宽，如果某规格的带宽为10 MB/s，则该规格实例的上下行带宽都是10 MB/s。
- 表中的带宽为Redis实例的内网带宽。外网带宽取决于内网带宽，同时受到Redis实例与客户端之间的网络带宽限制，建议使用内网连接方式，排除外网影响，发挥最大的带宽性能。

企业版（性能增强型）

标准版-双副本

规格	InstanceClass（API 使用）	IO线程数	每秒连接数	最大连接数	总带宽（MB/s）
1G主从性能增强版	redis.amber.master.small.multithread	6	10000	30000	96
2G主从性能增强版	redis.amber.master.mid.multithread	6	10000	30000	96
4G主从性能增强版	redis.amber.master.standard.multithread	6	10000	30000	96
8G主从性能增强版	redis.amber.master.large.multithread	6	10000	30000	96
16G主从性能增强版	redis.amber.master.2xlarge.multithread	6	10000	30000	96
32G主从性能增强版	redis.amber.master.4xlarge.multithread	6	10000	30000	96
64G主从性能增强版	redis.amber.master.8xlarge.multithread	6	10000	30000	96

集群版

规格	InstanceClass（API 使用）	IO线程数	分片数	代理数	每秒连接数	最大连接数	每分片带宽	总带宽（MB/s）
2G集群性能增强版（2节点）	redis.amber.logic.sharding.1g.2db.0rodb.6proxy.multithread	6	2	4	50000	60000	96	192
4G集群性能增强版（2节点）	redis.amber.logic.sharding.2g.2db.0rodb.6proxy.multithread	6	2	4	50000	50000	96	192
8G集群性能增强版（4节点）	redis.amber.logic.sharding.2g.4db.0rodb.12proxy.multithread	6	4	4	50000	120000	96	384
16G集群性能增强版（8节点）	redis.amber.logic.sharding.2g.8db.0rodb.24proxy.multithread	6	8	8	50000	240000	96	768
32G集群性能增强版（8节点）	redis.amber.logic.sharding.4g.8db.0rodb.24proxy.multithread	6	8	8	50000	240000	96	768

规格	InstanceClass（API 使用）	IO线程数	分片数	代理数	每秒连接数	最大连接数	每分片带宽	总带宽（MB/s）
64G集群性能增强版（8节点）	redis.amber.logic.sharding.8g.8db.0rodb.24proxy.multithread	6	8	8	50000	240000	96	768
32G集群性能增强版（16节点）	redis.amber.logic.sharding.2g.16db.0rodb.48proxy.multithread	6	16	16	50000	480000	96	1536
64G集群性能增强版（16节点）	redis.amber.logic.sharding.4g.16db.0rodb.48proxy.multithread	6	16	16	50000	480000	96	1536
128G集群性能增强版（16节点）	redis.amber.logic.sharding.8g.16db.0rodb.48proxy.multithread	6	16	16	50000	480000	96	1536
256G集群性能增强版（16节点）	redis.amber.logic.sharding.16g.16db.0rodb.48proxy.multithread	6	16	16	50000	480000	96	1536
512G集群性能增强版（32节点）	redis.amber.logic.sharding.16g.32db.0rodb.96proxy.multithread	6	32	32	50000	500000	96	2048
1024G集群性能增强版（64节点）	redis.amber.logic.sharding.16g.64db.0rodb.192proxy.multithread	6	64	64	50000	500000	96	2048
2048G集群性能增强版（128节点）	redis.amber.logic.sharding.16g.128db.0rodb.384proxy.multithread	6	128	128	50000	500000	96	2048
4096G集群性能增强版（256节点）	redis.amber.logic.sharding.16g.256db.0rodb.768proxy.multithread	6	256	256	50000	500000	96	2048
8192G集群性能增强版（256节点）	redis.amber.logic.sharding.32g.256db.0rodb.768proxy.multithread	6	256	256	50000	500000	96	2048

社区版

标准版-双副本

规格	InstanceClass（API 使用）	连接数上限	总带宽（MB/s）	处理能力
1G主从版	redis.master.small.default	10000	10	单核

规格	InstanceClass（API 使用）	连接数上限	总带宽（MB/s）	处理能力
2G主从版	redis.master.mid.default	10000	16	单核
4G主从版	redis.master.stand.default	10000	24	单核
8G主从版	redis.master.large.default	10000	24	单核
16G主从版	redis.master.2xlarge.default	10000	32	单核
32G主从版	redis.master.4xlarge.default	10000	32	单核
64G主从版	redis.master.8xlarge.default	10000	48	单核

集群版

规格	InstanceClass（API 使用）	分片数	连接数上限	总带宽（MB/s）	处理能力
16G集群版	redis.logic.sharding.2g.8db.0rodb.8proxy.default	8	80000	768	8核
32G集群版	redis.logic.sharding.4g.8db.0rodb.8proxy.default	8	80000	768	8核
64G集群版	redis.logic.sharding.8g.8db.0rodb.8proxy.default	8	80000	768	8核
128G集群版	redis.logic.sharding.8g.16db.0rodb.16proxy.default	16	160000	1536	16核
256G集群版	redis.logic.sharding.16g.16db.0rodb.16proxy.default	16	160000	1536	16核
512G集群版	redis.logic.sharding.16g.32db.0rodb.32proxy.default	32	320000	2048	32核
1T集群版	redis.logic.sharding.16g.64db.0rodb.64proxy.default	64	500000	2048	64核
2T集群版	redis.logic.sharding.16g.128db.0rodb.128proxy.default	128	500000	2048	128核
4T集群版	redis.logic.sharding.16g.256db.0rodb.256proxy.default	256	500000	2048	256核

读写分离版

规格	InstanceClass（API 使用）	只读节点数	每秒连接数	最大连接数	总带宽	QPS
1G读写分离版（1节点1只读）	redis.logic.splitrw.small.1db.1rodb.4proxy.default	1	20000	20000	96	200000
1G读写分离版（1节点3只读）	redis.logic.splitrw.small.1db.3rodb.4proxy.default	3	40000	40000	192	400000
2G读写分离版（1节点1只读）	redis.logic.splitrw.mid.1db.1rodb.4proxy.default	1	20000	20000	192	200000
2G读写分离版（1节点3只读）	redis.logic.splitrw.mid.1db.3rodb.4proxy.default	3	40000	40000	384	400000
4G读写分离版（1节点1只读）	redis.logic.splitrw.standard.1db.1rodb.4proxy.default	1	20000	20000	192	200000
4G读写分离版（1节点3只读）	redis.logic.splitrw.standard.1db.3rodb.4proxy.default	3	40000	40000	384	400000
8G读写分离版（1节点1只读）	redis.logic.splitrw.large.1db.1rodb.4proxy.default	1	20000	20000	192	200000
8G读写分离版（1节点3只读）	redis.logic.splitrw.large.1db.3rodb.4proxy.default	3	40000	40000	384	400000
16G读写分离版（1节点1只读）	redis.logic.splitrw.2xlarge.1db.1rodb.4proxy.default	1	20000	20000	192	200000
16G读写分离版（1节点3只读）	redis.logic.splitrw.2xlarge.1db.3rodb.4proxy.default	3	40000	40000	384	400000
32G读写分离版（1节点1只读）	redis.logic.splitrw.4xlarge.1db.1rodb.4proxy.default	1	20000	20000	192	200000
32G读写分离版（1节点3只读）	redis.logic.splitrw.4xlarge.1db.3rodb.4proxy.default	3	40000	40000	384	400000
64G读写分离版（1节点1只读）	redis.logic.splitrw.8xlarge.1db.1rodb.4proxy.default	1	20000	20000	192	200000

规格	InstanceClass（API 使用）	只读节点数	每秒连接数	最大连接数	总带宽	QPS
64G读写分离版（1节点3只读）	redis.logic.splitrw.8xlarge.1db.3rodb.4proxy.default	3	40000	40000	384	400000

2.12.2. 错误码表

错误码描述	Code	message	httpStatusCode
子账号调用 API 未经授权	Forbidden.RAM	User not authorized to operate on the specified resource, or this API doesn't support RAM.	403
该操作不支持 RAM 模式访问	Forbidden.NotSupportRAM	This action does not support accessed by RAM mode.	403
服务器端异常错误	ServiceUnavailable	The request has failed due to a temporary failure of the server.	503
输入的实例状态不存在	InvalidStatus.NotFound	The Status provided does not exist in our records.	404
输入的参数不合法	InvalidParameter	The specified parameter InstanceName is not valid.	400
普通用户调用管理类 API	Forbidden.NotAdminUser	User not authorized to operate on the specified API as you are not the administrator.	403
缺少参数	MissingParameter	Specified parameter is missing.	400
至少需要包含 InstanceName 或 NewPassword 参数两者之一	MissingParameter	InstanceName/NewPassword at least one is mandatory for this action.	400
调用该接口没有指定 OwnerId	MissingParameter	The input parameter OwnerId, OwnerAccount that is mandatory for processing this request is not supplied.	403

错误码描述	Code	message	httpStatusCode
指定的参数 Token 不合法	InvalidToken.Malformed	The Specified parameter "Token" is not valid.	400
指定的参数 InstanceName 不合法	InvalidInstanceName.Malformed	The Specified parameter "InstanceName" is not valid.	400
指定的参数 Password 不合法	InvalidPassword.Malformed	The Specified parameter "Password" is not valid."	400
指定的参数 Instances 不合法	InvalidInstances.Malformed	The Specified parameter "Instances" is not valid.	400
指定的参数 StartTime 不合法	InvalidStartTime.Malformed	The Specified parameter "StartTime" is not valid.	400
指定的参数 EndTime 不合法	InvalidEndTime.Malformed	The Specified parameter "EndTime" is not valid.	400
指定的参数 InstanceIds 不合法	InvalidInstanceIds.Malformed	The Specified parameter "InstanceIds" is not valid.	400
余额不足	InsufficientBalance	Your account does not have enough balance.	400
用户未进行实名认证	RealNameAuthenticationError	Your account has not passed the real-name authentication yet.	403
购买数量超出限制	QuotaExceed	Living afterpay instances quota exceeded.	400
容量配置不合法	InvalidCapacity.NotFound	The Capacity provided does not exist in our records.	400
用了一个已经使用过的 ClientToken，但此次请求内容却又与上一次使用该 Token 的请求内容又不一样	IdempotentParameterMismatch	Request uses a client token in a previous request but is not identical to that request.	400
指定的可用区库存不足	QuotaNotEnough	Quota not enough in this zone.	400

错误码描述	Code	message	httpStatusCode
无权调用订单类接口	Forbidden.SubUser	The specified action is not available for you.	403
阿里云风控系统拒绝了此次访问	Forbidden.RiskControl	This operation is forbidden by Aliyun Risk Control system.	403
指定的 Region 不存在	InvalidRegion.NotFound	The RegionId or ZoneId provided does not exist in our records.	404
指定的 ZoneId 不合法	InvalidZoneId.NotFound	The ZoneId provided is invalid.	400
实例 ID 不存在	InvalidInstanceId.NotFound	The InstanceId provided does not exist in our records.	404
密码不正确	IncorrectPassword	The Password provided is not correct.	400
服务不可用	ServiceNotSupported	The specified service is not supported.	400
有未生效的续费变配订单	HasRenewChangeOrder	This instance has a renewChange order.	400
内部错误	InternalServerError	The request processing has failed due to some unknown error.	500
备份时间无效	InvalidPreferredBackupTime	Specified preferred backup time is not valid.	400
输入备份类型不合法	InvalidBackupType.Format	Specified backup type is not valid.	400
输入的备份方法不合法	InvalidBackupMethod.Format	Specified backup method is not valid.	400
备份任务已经存在，暂不能支持	BackupJobExists	A backup job already exists in the specified DB instance.	400
备份次数已经超出限制	BackupTimesExceeded	Exceeding the daily backup times of this DB instance.	400
必须输入一个参数	ParameterLeastAssociate	Must input at least one optional parameter.	400

错误码描述	Code	message	httpStatusCode
输入的备份周期不合法	InvalidBackupRetentionPeriod.Malformed	Specified backup retention period is not valid.	400
输入的下次备份时间不合法	InvalidPreferredBackupTime.Format	Specified preferred backup time is not valid.	400
输入的备份周期不合法	InvalidPreferredBackupPeriod.Malformed	Specified backup period is not valid.	400
当前的实例类型不支持此操作	IncorrectDBInstanceType	Current DB instance type does not support this operation	400
输入的 key 不合法	InvalidKey.Malformed	Specified key is not valid.	400
签名已经被使用	SignatureNonceUsed	Specified signature nonce was used already.	400
已经无法分配虚拟 ip	AllocateVpcIp.NotEnough	Ip resource is not enough	400
该可用区不支持创建此类型实例	Zone.NotSupport	Specified zone does not support creating with instance class.	400
规格代码不合法	MissingClassCode	Capacity or InstanceClass is mandatory for this action.	400
实例规格不支持	InvalidDBInstanceClass.NotFound	Specified DB instance class is not found.	404
实例已经被锁定	IncorrectDBInstanceLockMode	Current DB instance lock mode does not support this operation.	400
备份集 id 不存在	InvalidBackupSetID.NotFound	Specified backup set ID does not exist.	400
实例状态不支持此操作	IncorrectDBInstanceState	Current DB instance state does not support this operation.	400
资源不足	InsufficientResourceCapacity	There is insufficient capacity available for the requested instance.	400
输入的结束时间无效	InvalidEndTime.Format	Specified end time is not valid.	400

错误码描述	Code	message	httpStatusCode
输入的保留经典 ip 的时间长度不合法	InvalidClassicExpiredDays.Format	The specified classicExpiredDays format is not valid.	400
备份集状态不支持此操作	IncorrectBackupSetState	Current backup set state does not support operations.	400
白名单格式不合法	InvalidSecurityIPList.Format	Specified security IP list format is not valid.	400

2.12.3. 实例配置参数表

Redis 实例参数配置表

参数名称	参数含义	参数值
maxmemory-policy	内存超过阈值时的逐出策略，云数据库KVStore for Memcache支持 6 种数据逐出策略。	<ul style="list-style-type: none">• VolatileLRU：按照 LRU 算法逐出原有数据，但仅逐出设置了过期时间的数据。• VolatileTTL：仅逐出设置了过期时间的数据，并且是按照 TTL 由小到大的顺序进行逐出。• AllKeysLRU：按照 LRU 算法逐出原有数据。• VolatileRandom：随机逐出原有数据，但仅逐出设置了过期时间的数据。• AllKeysRandom：随机逐出原有数据。• NoEviction：不逐出任何数据，新数据的写入会得到一个错误信息。
hash-max-ziplist-entries	<p>当 hash 对象同时满足以下两个条件时，hash 对象使用 ziplist 编码：</p> <ul style="list-style-type: none">• hash 对象保存的键值对数量小于或者等于 list-max-ziplist-entries 参数值。• hash 对象保存的所有键值对的键和值的字符串长度都小于或者等于 list-max-ziplist-value 参数值。	默认值：512

参数名称	参数含义	参数值
hash-max-ziplist-value	当 hash 对象同时满足以下两个条件时，hash 对象使用 ziplist 编码： <ul style="list-style-type: none">hash 对象保存的键值对数量小于或者等于 list-max-ziplist-entries 参数值。hash 对象保存的所有键值对的键和值的字符串长度都小于或者等于 list-max-ziplist-value 参数值。	默认值：64
list-max-ziplist-entries	当 list 对象同时满足以下两个条件时，list 对象使用 ziplist 编码： <ul style="list-style-type: none">list 对象保存的键值对数量小于或者等于 list-max-ziplist-entries 参数值。list 对象保存的所有键值对的键和值的字符串长度都小于或者等于 list-max-ziplist-value 参数值。	默认值：512
list-max-ziplist-value	当 list 对象同时满足以下两个条件时，list 对象使用 ziplist 编码： <ul style="list-style-type: none">list 对象保存的键值对数量小于或者等于 list-max-ziplist-entries 参数值。list 对象保存的所有键值对的键和值的字符串长度都小于或者等于 list-max-ziplist-value 参数值。	默认值：64
set-max-intset-entries	当 set 对象满足条目小于或者等于 set-max-intset-entries 参数值并且都是十进制整数时，编码为 intset。	默认值：512
zset-max-ziplist-entries	当 zset 对象同时满足以下两个条件时，zset 对象使用 ziplist 编码： <ul style="list-style-type: none">zset 对象保存的键值对数量小于或者等于 zset-max-ziplist-entries 参数值。zset 对象保存的所有键值对的键和值的字符串长度都小于或者等于 zset-max-ziplist-value 参数值。	默认值：128

参数名称	参数含义	参数值
zset-max-ziplist-value	<p>当 zset 对象同时满足以下两个条件时，zset 对象使用 ziplist 编码：</p> <ul style="list-style-type: none">• zset 对象保存的键值对数量小于或者等于 zset-max-ziplist-entries 参数值。• zset 对象保存的所有键值对的键和值的字符串长度都小于或者等于 zset-max-ziplist-value 参数值。	默认值：64
notify-keyspace-events	键空间通知使得客户端可以通过订阅频道或模式来接收那些以某种方式改动了 Redis 数据集的事件。	<ul style="list-style-type: none">• K 键空间通知，所有通知以 <code>__keyspace@__</code> 为前缀。• E 键事件通知，所有通知以 <code>__keyevent@__</code> 为前缀。• g DEL 、 EXPIRE 、 RENAME 等类型无关的通用命令的通知。• \$ 字符串命令的通知。• l 列表命令的通知。• s 集合命令的通知。• h 哈希命令的通知。• z 有序集合命令的通知。• x 过期事件：每当有过期键被删除时发送。• e 驱逐（evict）事件：每当有键因为 maxmemory 政策而被删除时发送。• A 参数 g\$shzxe 的别名。