

ALIBABA CLOUD

# Alibaba Cloud

消息队列Kafka版  
API参考

文档版本：20220624

 阿里云

## 法律声明

阿里云提醒您在使用或阅读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。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击 <b>确定</b> 。
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> <i>Instance_ID</i>
[ ] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1.API概览	06
2.API版本说明	08
3.调用方式	09
4.获取SDK	10
5.服务接入点	11
6.请求示例	12
7.签名机制	17
8.公共参数	20
9.获取AccessKey	22
10.实例	24
10.1. CreatePostPayOrder	24
10.2. CreatePrePayOrder	26
10.3. StartInstance	29
10.4. UpgradePostPayOrder	34
10.5. UpgradePrePayOrder	37
10.6. ConvertPostPayOrder	39
10.7. ModifyInstanceName	41
10.8. ReleaseInstance	43
10.9. GetInstanceList	45
10.10. GetAllInstanceIdList	53
10.11. DeleteInstance	54
10.12. GetAllowedIpList	56
10.13. UpdateAllowedIp	59
10.14. UpdateInstanceConfig	61
10.15. UpgradeInstanceVersion	63
11.Topic	67

---

11.1. CreateTopic	67
11.2. DeleteTopic	70
11.3. GetTopicList	71
11.4. GetTopicStatus	75
11.5. ModifyPartitionNum	79
11.6. ModifyTopicRemark	81
12.Group	83
12.1. DeleteConsumerGroup	83
12.2. GetConsumerList	84
12.3. GetConsumerProgress	87
12.4. CreateConsumerGroup	90
13.标签	93
13.1. UntagResources	93
13.2. TagResources	95
13.3. ListTagResources	97
14.SASL用户	101
14.1. CreateSaslUser	101
14.2. DeleteSaslUser	102
14.3. DescribeSaslUsers	104
15.SASL ACL	107
15.1. CreateAcl	107
15.2. DescribeAcls	109
15.3. DeleteAcl	112

# 1.API概览

本文列举消息队列Kafka版支持的API接口。

## 实例

API	描述
<a href="#">GetInstanceList</a>	调用GetInstanceList获取某地域下所购买实例的信息。
<a href="#">StartInstance</a>	调用StartInstance部署实例。
<a href="#">ModifyInstanceName</a>	调用ModifyInstanceName修改实例的别名。
<a href="#">CreatePrePayOrder</a>	调用CreatePrePayOrder创建预付费实例。
<a href="#">DeleteInstance</a>	调用DeleteInstance删除实例。
<a href="#">UpgradePrePayOrder</a>	调用UpgradePrePayOrder升配预付费实例。
<a href="#">GetAllowedIpList</a>	调用GetAllowedIpList获取IP白名单。
<a href="#">UpdateAllowedIp</a>	调用UpdateAllowedIp修改IP白名单。

## Topic

API	描述
<a href="#">CreateTopic</a>	调用CreateTopic创建Topic。
<a href="#">DeleteTopic</a>	调用DeleteTopic删除Topic。
<a href="#">GetTopicList</a>	调用GetTopicList获取Topic的列表。
<a href="#">GetTopicStatus</a>	调用GetTopicStatus获取Topic的消息收发状态。
<a href="#">ModifyPartitionNum</a>	调用ModifyPartitionNum修改Topic的分区数。
<a href="#">ModifyTopicRemark</a>	调用ModifyTopicRemark修改Topic的备注。

## Group

API	描述
<a href="#">CreateConsumerGroup</a>	调用CreateConsumerGroup创建Group。
<a href="#">DeleteConsumerGroup</a>	调用DeleteConsumerGroup删除Group。
<a href="#">GetConsumerList</a>	调用GetConsumerList获取Group的列表。
<a href="#">GetConsumerProgress</a>	调用GetConsumerProgress查询Group的消费状态。

## 标签

API	描述
ListTagResources	调用ListTagResources查询资源绑定的标签信息。
TagResources	调用TagResources为资源创建并绑定标签。
UntagResources	调用UntagResources为资源解绑并删除标签。

## SASL ACL

API	描述
DescribeAcls	调用DescribeAcls查询ACL。
CreateAcl	调用CreateAcl创建ACL。
DeleteAcl	调用DeleteAcl删除ACL。

## SASL用户

API	描述
DescribeSaslUsers	调用DescribeSaslUsers查询SASL用户。
CreateSaslUser	调用CreateSaslUser创建SASL用户。
DeleteSaslUser	调用DeleteSaslUser删除SASL用户。

## 2.API版本说明

本文介绍消息队列Kafka版所有API版本的发布信息。

API版本	发布日期	说明
V1.0.0	2020年3月31日	<p>首次发布。支持以下接口：</p> <ul style="list-style-type: none"><li>• 实例<ul style="list-style-type: none"><li>◦ <a href="#">CreatePrePayOrder</a></li><li>◦ <a href="#">StartInstance</a></li><li>◦ <a href="#">UpgradePrePayOrder</a></li><li>◦ <a href="#">ModifyInstanceName</a></li><li>◦ <a href="#">DeleteInstance</a></li><li>◦ <a href="#">GetAllowedIpList</a></li><li>◦ <a href="#">UpdateAllowedIp</a></li><li>◦ <a href="#">GetInstanceList</a></li></ul></li><li>• Topic<ul style="list-style-type: none"><li>◦ <a href="#">CreateTopic</a></li><li>◦ <a href="#">DeleteTopic</a></li><li>◦ <a href="#">GetTopicList</a></li><li>◦ <a href="#">GetTopicStatus</a></li></ul></li><li>• Consumer Group<ul style="list-style-type: none"><li>◦ <a href="#">CreateConsumerGroup</a></li><li>◦ <a href="#">GetConsumerList</a></li><li>◦ <a href="#">DeleteConsumerGroup</a></li><li>◦ <a href="#">GetConsumerProgress</a></li></ul></li><li>• Tag<ul style="list-style-type: none"><li>◦ <a href="#">ListTagResources</a></li><li>◦ <a href="#">TagResources</a></li><li>◦ <a href="#">UntagResources</a></li></ul></li></ul>



## 3.调用方式

消息队列Kafka版支持CommonRequest调用、HTTP调用、SDK调用和OpenAPI Explorer调用。

### CommonRequest调用

当您要调用的消息队列Kafka版的API没有及时更新SDK时，可以采用CommonRequest调用方式，实现任意OpenAPI接口的调用。消息队列Kafka版的API是RPC风格，您可以通过发送CommonRequest请求调用消息队列Kafka版API。

发起一次CommonRequest请求，您需要获取的参数值如下：

- Domain：实例所在服务地域。更多信息，请参见[服务接入点](#)。
- Version：要使用的API版本。消息队列Kafka版的API版本是2019-09-16。
- Action：要调用的接口名称。需要获取Action参数，使用 `request.ApiName = "<Action>"` 的方式来指定API名称。例如RunInstances接口，在发起CommonRequest请求时，需要使用 `request.ApiName = "RunInstances"` 来指定API名称。更多信息，请参见[API概览](#)。

### HTTP调用

消息队列Kafka版HTTP调用是向消息队列Kafka版的服务端地址发送HTTP请求，并按照接口说明在请求中加入相应请求参数，调用后系统会返回处理结果。请求及返回结果都使用UTF-8字符集进行编码消息队列Kafka版的API是RPC风格，您可以通过发送HTTP请求调用消息队列Kafka版API。

其请求结构如下：

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

其中：

- Endpoint：消息队列Kafka版API的服务接入地址。更多信息，请参见[服务接入点](#)。
- Action：要执行的操作。例如，调用GetInstanceList查询已创建的消息队列Kafka版实例。
- Version：要使用的API版本。消息队列Kafka版的API版本是2019-09-16。
- Parameters：请求参数。每个参数之间用“&”分隔。请求参数由公共请求参数和API自定义参数组成。公共参数中包含API版本号、身份验证等信息。更多信息，请参见[公共参数](#)。

### SDK调用

消息队列Kafka版提供多语言SDK。消息队列Kafka版SDK可以免去您手动签名的过程。更多信息，请参见[获取SDK](#)。

### OpenAPI Explorer调用

OpenAPI Explorer是可视化的API调用工具。通过该工具，您可以通过网页或者命令行调用各云产品以及API市场上开放的API，查看每次的API请求和返回结果，并生成相应SDK调用示例。您可以直接访问[OpenAPI Explorer](#)。

## 4. 获取SDK

本文介绍如何获取消息队列Kafka版的SDK。

消息队列Kafka版支持以下语言的SDK。

语言	SDK
Java	<a href="#">Aliyun Java SDK Alikafka</a>
Go	<a href="#">Aliyun Go SDK Alikafka</a>
PHP	<a href="#">Aliyun PHP SDK Alikafka</a>

## 5. 服务接入点

本文列出消息队列Kafka版的API接入地域ID和域名地址。

地域名称	RegionId	Domain
华东1（杭州）	cn-hangzhou	alikafka.cn-hangzhou.aliyuncs.com
华东2（上海）	cn-shanghai	alikafka.cn-shanghai.aliyuncs.com
华北1（青岛）	cn-qingdao	alikafka.cn-qingdao.aliyuncs.com
华北2（北京）	cn-beijing	alikafka.cn-beijing.aliyuncs.com
华北3（张家口）	cn-zhangjiakou	alikafka.cn-zhangjiakou.aliyuncs.com
华北5（呼和浩特）	cn-huhehaote	alikafka.cn-huhehaote.aliyuncs.com
华北6（乌兰察布）	cn-wulanchabu	alikafka.cn-wulanchabu.aliyuncs.com
华南1（深圳）	cn-shenzhen	alikafka.cn-shenzhen.aliyuncs.com
华南2（河源）	cn-heyuan	alikafka.cn-heyuan.aliyuncs.com
华南3（广州）	cn-guangzhou	alikafka.cn-guangzhou.aliyuncs.com
西南1（成都）	cn-chengdu	alikafka.cn-chengdu.aliyuncs.com
中国香港	cn-hongkong	alikafka.cn-hongkong.aliyuncs.com
新加坡（新加坡）	ap-southeast-1	alikafka.ap-southeast-1.aliyuncs.com
马来西亚（吉隆坡）	ap-southeast-3	alikafka.ap-southeast-3.aliyuncs.com
印度尼西亚（雅加达）	ap-southeast-5	alikafka.ap-southeast-5.aliyuncs.com
日本（东京）	ap-northeast-1	alikafka.ap-northeast-1.aliyuncs.com
德国（法兰克福）	eu-central-1	alikafka.eu-central-1.aliyuncs.com
英国（伦敦）	eu-west-1	alikafka.eu-west-1.aliyuncs.com
美国（硅谷）	us-west-1	alikafka.us-west-1.aliyuncs.com
美国（弗吉尼亚）	us-east-1	alikafka.us-east-1.aliyuncs.com
印度（孟买）	ap-south-1	alikafka.ap-south-1.aliyuncs.com
菲律宾（马尼拉）	ap-southeast-6	alikafka.ap-southeast-6.aliyuncs.com

## 6. 请求示例

本文提供使用Java语言构造参数和签名并POST发送请求的示例，其他语言流程类似。

### CommonRequest调用使用示例

```
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.profile.DefaultProfile;
/**
 * Demo所需依赖为Alibaba Cloud SDK for Java。
 * 您可以通过直接添加Maven依赖或者下载Alibaba Cloud SDK for Java的开发工具包的方式安装Alibaba Cloud SDK for Java。
 *
 * 本Demo为泛用型的RPC风格的API调用方式。
 * 使用CommonRequest调用方式可实现任意OpenAPI接口的调用。
 * Demo调用了消息队列Kafka版的GetTopicList（获取Topic信息）接口。
 */
public class CommonRequestDemo {
    // 购买的实例所在RegionId。
    private static final String REGION_ID = "cn-huhehaote";
    // 您的AccessKey ID。
    private static final String ACCESS_KEY = "LTAI5tHRxaQd8WeJvDFZ****";
    // 您的AccessKey Secret。
    private static final String ACCESS_KEY_SECRET = "R1WUTHmTD1Cy7J8csQ6bRnaMkY****";
    // 实例所在服务地域。
    private static final String SYS_DO_MAIN = "alikafka.cn-huhehaote.aliyuncs.com";
    public static void main(String[] args) {
        // 创建DefaultAcsClient实例并初始化
        DefaultProfile profile = DefaultProfile.getProfile(
            REGION_ID,          // 您的地域ID。
            ACCESS_KEY,        // 您的AccessKey ID。
            ACCESS_KEY_SECRET); // 您的AccessKey Secret。
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建API请求并设置参数。
        CommonRequest request = new CommonRequest();
        // 实例所在服务地域。
        request.setSysDomain(SYS_DO_MAIN);
        // API版本。
        request.setSysVersion("2019-09-16");
        // 接口名称。
        request.setSysAction("GetTopicList");
        // 接口参数。
        request.putQueryParameter("InstanceId", "alikafka_pre-cn-tl32d052****");
        request.putQueryParameter("PageNumber", "1");
        request.putQueryParameter("PageSize", "30");
        try {
            CommonResponse response = client.getCommonResponse(request);
            System.out.println(response.getData());
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

## HTTP调用使用示例

```
private static final String ISO8601_DATE_FORMAT = "yyyy-MM-dd'T'HH:mm:ss'Z'";
private static final String ENCODING = "UTF-8";
private static final String HTTP_METHOD = "POST";
// 消息队列Kafka版实例的地域的Region ID。
private static final String REGION_ID = "cn-xxxxxxx";
private static final String ACCESS_KEY = "xxxxxxx";
private static final String ACCESS_KEY_SECRET = "xxxxxxx";
public static void main(String[] args) {
    try {
        // 1. 设置参数。
        Map<String, String> parameters = buildCommonParams();
        // 2. 排序请求参数：根据字母排序。
        String[] sortedKeys = sortParamsToArray(parameters);
        // 3. 构造stringToSign字符串。
        String stringToSign = buildSignString(parameters, sortedKeys);
        // 4. 计算签名。
        String signature = calculateSignature(stringToSign);
        // 5. 将签名设置到参数中。
        parameters.put("Signature", signature);
        // 6. 发送POST请求。
        String result = post(String.format("http://alikafka.%s.aliyuncs.com/", REGION_ID), parameters);
        // 7. 验证结果。
        System.out.println(result);
    } catch (Throwable throwable) {
        throwable.printStackTrace();
    }
}

private static Map<String, String> buildCommonParams() {
    Map<String, String> parameters = Maps.newHashMap();
    // Action为请求方法。
    // GetInstanceList: 获取实例信息; GetConsumerList: 获取消费组列表; GetTopicList: 获取Topic列表; GetTopicStatus: 获取 Topic 状态。
    // GetConsumerProgress: 获取消费状态; CreateTopic: 创建Topic; CreateConsumerGroup: 创建Group。
    parameters.put("Action", "GetInstanceList");
    // 接口版本: "2018-10-15"。
    parameters.put("Version", "2018-10-15");
    // 请求参数: RegionId为必填参数。
    // 如果接口有其他参数请都设置: 比如 InstanceId/Topic/ConsumerId。
    // parameters.put("InstanceId", "cn-huhehaote");
    // parameters.put("Topic", "cn-huhehaote");
    // parameters.put("Remark", "cn-huhehaote");
    // parameters.put("ConsumerId", "cn-huhehaote");
    parameters.put("RegionId", REGION_ID);
    parameters.put("AccessKeyId", ACCESS_KEY);
    // 时间戳, 注意格式: yyyy-MM-dd'T'HH:mm:ss'Z'。
    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");
}
```

```
        return parameters;
    }
    private static String[] sortParamsToArray(Map<String, String> parameters) {
        String[] sortedKeys = parameters.keySet().toArray(new String[]{});
        Arrays.sort(sortedKeys);
        return sortedKeys;
    }
    private static String buildSignString(Map<String, String> parameters,
        String[] sortedKeys) throws UnsupportedEncodingException {
        StringBuilder stringToSign = new StringBuilder();
        String SEPARATOR = "&";
        stringToSign.append(HTTP_METHOD).append(SEPARATOR);
        stringToSign.append(percentEncode("/")).append(SEPARATOR);
        StringBuilder canonicalizedQueryString = new StringBuilder();
        for(String key : sortedKeys) {
            // 注意编码key和value。
            canonicalizedQueryString.append("&")
                .append(percentEncode(key)).append("=")
                .append(percentEncode(parameters.get(key)));
        }
        // 注意编码canonicalizedQueryString。
        stringToSign.append(percentEncode(canonicalizedQueryString.toString().substring(1)));
    };
    return stringToSign.toString();
}
private static String calculateSignature(String stringToSign)
    throws NoSuchAlgorithmException, InvalidKeyException, UnsupportedEncodingException
{
    String ALGORITHM = "HmacSHA1";
    String ENCODING = "UTF-8";
    // 对应账号的AccessKeySecret。
    String accessKeySecret = ACCESS_KEY_SECRET + "&";
    Mac mac = Mac.getInstance(ALGORITHM);
    mac.init(new SecretKeySpec(accessKeySecret.getBytes(ENCODING), ALGORITHM));
    byte[] signData = mac.doFinal(stringToSign.getBytes(ENCODING));
    return new String(Base64.getEncoder().encode(signData));
}
private static String percentEncode(String value) throws UnsupportedEncodingException {
    return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace(
        "*", "%2A").replace("%7E", "~") : null;
}
private static String formatIso8601Date(Date date) {
    SimpleDateFormat df = new SimpleDateFormat(ISO8601_DATE_FORMAT);
    df.setTimeZone(new SimpleTimeZone(0, "GMT"));
    return df.format(date);
}
private static String post(String url, Map<String, String> paramMap) throws IOException
{
    Form form = Form.form();
    for (String key : paramMap.keySet()) {
        form.add(key, paramMap.get(key));
    }
    return Request.Post(url).bodyForm(form.build()).connectTimeout(10000).execute().ret
```


```
urnContent().asString();  
}
```



## 7. 签名机制

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

您的用户名和密码是用于登录消息队列Kafka版控制台的身份凭据，与此类似，AccessKey是用于调用API的身份凭据。AccessKey中的AccessKeyId是访问者身份，AccessKeySecret是加密签名字符串和服务端验证签名字符串的密钥，必须严格保密谨防泄露。

 **说明** 消息队列Kafka版提供了多种编程语言的SDK，可以免去您签名的烦恼。具体信息，请参见[SDK](#)。

### 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. 编码参数。使用UTF-8字符集按照RFC3986规则编码请求参数和参数取值，编码规则如下：
  - 字符A~Z、a~z、0~9以及符号短划线 (-)、下划线 (\_)、半角句号 (.)、波浪线 (~) 不编码。
  - 其它字符编码成 `%XY` 的格式，其中 `XY` 是字符对应ASCII码的十六进制。示例：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` 替换为波浪线 (~)，即可得到上述规则描述的编码字符串。示例如下：

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

3. 使用等于号 (=) 连接编码后的请求参数和参数取值。
4. 使用and (&) 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[HTTP调用](#)。

### 步骤二：构造签名字符串

1. 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POST。percentEncode("/") + "&" +
//percentEncode("/") : 字符 (/) UTF-8编码得到的值, 即 %2F。percentEncode(CanonicalizedQueryS
tring) //您的规范化请求字符串。
```

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

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

**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。

- 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 `GetInstanceList` 获取实例列表为例。假设您获得了 `AccessKeyId=testid` 以及 `AccessKeySecret=testsecret`，签名流程如下所示：

- 构造规范化请求字符串。

```
http://alikafka.%s.aliyuncs.com/?Timestamp=2016-02-23T12:46:24Z&Format=XML&AccessKeyId=testid&Action=GetInstanceList&SignatureMethod=HMAC-SHA1&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf&Version=2014-05-26&SignatureVersion=1.0
```

- 构造待签名字符串 `StringToSign`。

```
POST%2F&AccessKeyId%3Dtestid&Action%3DGetInstanceList&Format%3DXML&SignatureMethod%3DHMAC-SHA1&SignatureNonce%3D3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf&SignatureVersion%3D1.0&Timestamp%3D2016-02-23T12%253A46%253A24Z&Version%3D2014-05-26
```

- 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 `OLeaidS1JvxuMvnyHOwuJ+uX5qY=`。示例使用的是Java Base64编码方法。

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

- 添加RFC3986规则编码后的 `Signature=OLeaidS1JvxuMvnyHOwuJ%2BuX5qY%3D` 到步骤1的URL中。

```
http://alikafka.%s.aliyuncs.com/?SignatureVersion=1.0&Action=GetInstanceList&Format=JSON&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf&Version=2014-05-26&AccessKeyId=testid&Signature=OLeaidS1JvxuMvnyHOwuJ%2BuX5qY%3D&SignatureMethod=HMAC-SHA1&Timestamp=2016-02-23T12%253A46%253A24Z
```

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 `GetInstanceList`，查询指定地域的实例信息。

## 示例二：编程语言法

依然以调用 `GetInstanceList` 获取实例列表为例。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。

- 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 = "POST";
Map parameters = new HashMap();
// 输入请求参数。
parameters.put("Action", "GetInstanceList");
parameters.put("Version", "2014-05-26");
parameters.put("AccessKeyId", "testid");
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");
// 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&` ，计算得到的签名值为 `OLeaidS1JvxuMvnyHOwuJ%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 = new String(Base64.encodeBase64(signData));
```

## 8. 公共参数

本文列举了消息队列Kafka版的API的公共请求参数和公共返回参数。

### 公共请求参数

公共请求参数是每个接口都需要使用到的请求参数。

名称	类型	是否必需	描述
Format	String	否	返回消息的格式。取值：JSON   XML（默认值）
Version	String	是	API版本号，使用YYYY-MM-DD日期格式。取值：2019-09-16
AccessKeyId	String	是	访问服务使用的密钥ID。
Signature	String	是	签名结果串。
SignatureMethod	String	是	签名方式。取值：HMAC-SHA1
Timestamp	String	是	请求的时间戳，为日期格式。使用UTC时间按照ISO 8601标准，格式为YYYY-MM-DDThh:mm:ssZ。例如，北京时间2013年1月10日20点0分0秒，表示为2013-01-10T12:00:00Z。
SignatureVersion	String	是	签名算法版本。取值：1.0
SignatureNonce	String	是	唯一随机数，用于防止网络重放攻击。在不同请求间要使用不同的随机数值。

请求示例：

```
https://alikafka.cn-hangzhou.aliyuncs.com/?Action=GetInstanceList
&Format=JSON
&Version=2019-09-16
&AccessKeyId=key-test
&Signature=Pc5WB8gokVn0xfeu%2FZV%2BiNM1dgI%3D
&SignatureMethod=HMAC-SHA1
&Timestamp=2020-01-01T12:00:00Z
&SignatureNonce=15215528852396
&SignatureVersion=1.0
...
```

## 公共返回参数

每次接口调用请求，无论成功与否，系统都会返回一个唯一识别码RequestId。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；返回的HTTP状态码为 4xx 或 5xx ，代表调用失败。

XML示例：

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

JSON示例：

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


## 9. 获取AccessKey

您可以为阿里云主账号和子账号创建一个访问密钥（AccessKey）。在调用阿里云API时您需要使用AccessKey完成身份验证。

### 背景信息

AccessKey包括AccessKey ID和AccessKey Secret。

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

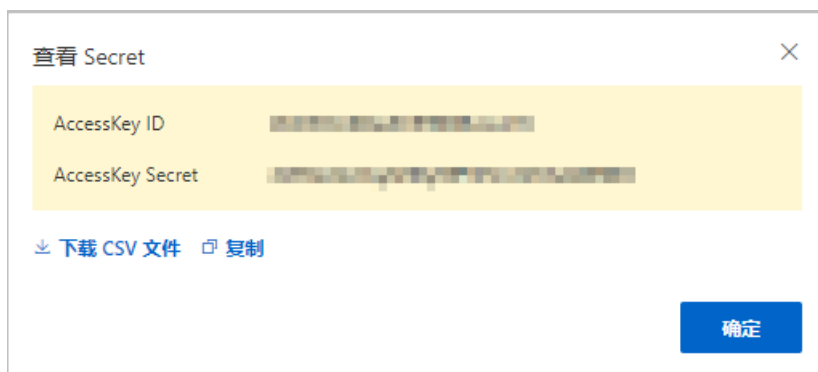
 **警告** 主账号AccessKey泄露会威胁您所有资源的安全。建议使用子账号（RAM用户）AccessKey进行操作，可以有效降低AccessKey泄露的风险。

### 操作步骤

1. 以主账号登录[阿里云管理控制台](#)。
2. 将鼠标置于页面右上方的账号图标，单击**AccessKey管理**。
3. 在**安全提示**页面，选择获取主账号还是子账号的AccessKey。



4. 获取账号AccessKey。
  - 获取主账号AccessKey
    - a. 单击**继续使用AccessKey**。
    - b. 在**安全信息管理**页面，单击**创建AccessKey**。
    - c. 在**手机验证**页面，获取验证码，完成手机验证，单击**确定**。
    - d. 在**新建用户AccessKey**页面，展开**AccessKey详情**，查看AccessKey ID和AccessKey Secret。可以单击**保存AK信息**，下载AccessKey信息。



- o 获取子账号AccessKey
  - a. 单击开始使用子用户AccessKey。
  - b. 如果未创建RAM用户，在系统跳转的RAM访问控制台的新建用户页面，创建RAM用户。如果是获取已有RAM用户的AccessKey，则跳过此步骤。
  - c. 在RAM访问控制台的左侧导航栏，选择人员管理 > 用户，搜索需要获取AccessKey的用户。
  - d. 单击用户登录名称，在用户详情页认证管理页签下的用户AccessKey区域，单击创建新的AccessKey
  - e. 在创建AccessKey页面，查看AccessKey ID和AccessKey Secret。可以单击下载CSV文件，下载AccessKey信息或者单击复制，复制AccessKey信息。



# 10. 实例

## 10.1. CreatePostPayOrder

调用CreatePostPayOrder创建后付费实例。

请确保在使用该接口前，已充分了解后付费实例的收费方式和价格。更多信息，请参见[计费说明](#)。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreatePostPayOrder	系统规定参数。取值： <b>CreatePostPayOrder</b> 。
RegionId	String	是	cn-hangzhou	实例的地域ID。
TopicQuota	Integer	是	50	Topic的数量。 <ul style="list-style-type: none"> <li>流量规格不同，默认值不同。超过默认值，需额外收费。</li> <li>取值范围，请参见<a href="#">计费说明</a>。</li> </ul>
DiskType	String	是	0	磁盘类型。取值： <ul style="list-style-type: none"> <li>0：高效云盘</li> <li>1：SSD</li> </ul>
DiskSize	Integer	是	500	磁盘容量。 <p>取值范围，请参见<a href="#">计费说明</a>。</p>
DeployType	Integer	是	5	部署类型。取值： <ul style="list-style-type: none"> <li>4：公网/VPC实例</li> <li>5：VPC实例</li> </ul>
IoMax	Integer	否	20	流量峰值（不推荐）。 <ul style="list-style-type: none"> <li>流量峰值和流量规格必须选填一个。同时填写时，以流量规格为准。建议您只填写流量规格。</li> <li>取值范围，请参见<a href="#">计费说明</a>。</li> </ul>



名称	类型	是否必选	示例值	描述
EipMax	Integer	否	0	公网流量。 <ul style="list-style-type: none"> <li>如果DeployType为4，则需填写。</li> <li>取值范围，请参见<a href="#">计费说明</a>。</li> </ul>
SpecType	String	否	normal	规格类型。取值： <ul style="list-style-type: none"> <li>normal：标准版（高写版）</li> <li>professional：专业版（高写版）</li> <li>professionalForHighRead：专业版（高读版）</li> </ul> 以上规格类型的说明，请参见 <a href="#">计费说明</a> 。
IoMaxSpec	String	否	alikafka.hw.2xlarge	流量规格（推荐）。 <ul style="list-style-type: none"> <li>流量峰值和流量规格必须选填一个。同时填写时，以流量规格为准。建议您只填写流量规格。</li> <li>取值范围，请参见<a href="#">计费说明</a>。</li> </ul>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	06084011-E093-46F3-A51F-4B19A8AD****	请求的ID。
Success	Boolean	true	调用是否成功。
OrderId	String	20497346575****	订单的ID。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=CreatePostPayOrder
&RegionId=cn-hangzhou
&TopicQuota=50
&DiskType=0
&DiskSize=500
&DeployType=5
&IoMax=20
&EipMax=0
&SpecType=normal
&IoMaxSpec=alibabacloud.com/hw.2xlarge
&公共请求参数
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<CreatePostPayOrderResponse>
  <Code>200</Code>
  <Message>operation success.</Message>
  <RequestId>06084011-E093-46F3-A51F-4B19A8AD****</RequestId>
  <Success>>true</Success>
  <OrderId>20497346575****</OrderId>
</CreatePostPayOrderResponse>
```

#### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : 200,
  "Message" : "operation success.",
  "RequestId" : "06084011-E093-46F3-A51F-4B19A8AD****",
  "Success" : true,
  "OrderId" : "20497346575****"
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 10.2. CreatePrePayOrder

调用CreatePrePayOrder创建预付费实例。

## 使用说明

- 请确保在使用该接口前，已充分了解预付费实例的收费方式和价格。更多信息，请参见[计费说明](#)。
- 使用该接口后，购买的预付费实例默认购买时长为一个月，并且默认开通了自动续费，自动续费周期为一个月。如果您需要修改自动续费周期或取消自动续费，您可以在阿里云控制台[续费管理](#)页面操作。更多信息，请参见[取消自动续费](#)。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreatePrePayOrder	系统规定参数。取值： <b>CreatePrePayOrder</b> 。
RegionId	String	是	cn-hangzhou	实例的地域ID。
TopicQuota	Integer	是	50	Topic的数量。 <ul style="list-style-type: none"> <li>流量规格不同，默认值不同。超过默认值，需额外收费。</li> <li>取值范围请参见<b>计费说明</b>。</li> </ul>
DiskType	String	是	1	磁盘类型。取值： <ul style="list-style-type: none"> <li>0：高效云盘</li> <li>1：SSD</li> </ul>
DiskSize	Integer	是	900	磁盘容量。 <p>取值范围请参见<b>计费说明</b>。</p>
DeployType	Integer	是	5	部署类型。取值： <ul style="list-style-type: none"> <li>4：公网/VPC实例</li> <li>5：VPC实例</li> </ul>
IoMax	Integer	否	20	流量峰值（不推荐）。 <ul style="list-style-type: none"> <li><b>IoMax</b>和<b>IoMaxSpec</b>必须至少填写一个。同时填写时，以<b>IoMaxSpec</b>为准。建议您只填写<b>IoMaxSpec</b>。</li> <li>取值范围请参见<b>计费说明</b>。</li> </ul>
EipMax	Integer	否	40	公网流量。 <ul style="list-style-type: none"> <li>如果<b>DeployType</b>为4，则需填写。</li> <li>取值范围请参见<b>按量计费</b>。</li> </ul>

名称	类型	是否必选	示例值	描述
SpecType	String	否	normal	规格类型。取值： <ul style="list-style-type: none"> <li>• <b>normal</b>：普通版（高写版）</li> <li>• <b>professional</b>：专业版（高写版）</li> <li>• <b>professionalForHighRead</b>：专业版（高读版）</li> </ul> 更多信息，请参见 <a href="#">计费说明</a> 。
IoMaxSpec	String	否	alikafka.hw.2xlarge	流量规格（推荐）。 <ul style="list-style-type: none"> <li>• <b>IoMax</b>和<b>IoMaxSpec</b>必须至少填写一个。同时填写时，以<b>IoMaxSpec</b>为准。建议您只填写<b>IoMaxSpec</b>。</li> <li>• 取值范围请参见<a href="#">计费说明</a>。</li> </ul>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	06084011-E093-46F3-A51F-4B19A8AD****	请求的ID。
Success	Boolean	true	调用是否成功。
OrderId	String	20497346575****	订单的ID。

## 示例

请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=CreatePrePayOrder
&RegionId=cn-hangzhou
&TopicQuota=50
&DiskType=1
&DiskSize=900
&DeployType=5
&IoMax=20
&EipMax=40
&SpecType=normal
&IoMaxSpec=alikafka.hw.2xlarge
&公共请求参数
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<CreatePrePayOrderResponse>
  <Code>200</Code>
  <Message>operation success.</Message>
  <RequestId>06084011-E093-46F3-A51F-4B19A8AD****</RequestId>
  <Success>>true</Success>
  <OrderId>20497346575****</OrderId>
</CreatePrePayOrderResponse>
```

#### JSON 格式


```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : 200,
  "Message" : "operation success.",
  "RequestId" : "06084011-E093-46F3-A51F-4B19A8AD****",
  "Success" : true,
  "OrderId" : "20497346575****"
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

## 10.3. StartInstance

调用StartInstance部署实例。

 **说明** 单用户请求频率限制为2 QPS。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	StartInstance	系统规定参数。取值： <b>StartInstance</b> 。
InstanceId	String	是	alikafka_post-cn-v0h1fgs2****	实例的ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。
VpcId	String	是	vpc-bp1r4eg3yrxmygv****	实例部署的VPC ID。
VSwitchId	String	是	vsw-bp1j3sg5979fstnpl****	实例部署的vSwitch ID。
DeployModule	String	是	vpc	部署模式。取值： <ul style="list-style-type: none"> <li><b>vpc</b>：VPC实例</li> <li><b>eip</b>：公网/VPC实例</li> </ul> 实例的部署模式必须与其类型一致。VPC实例，部署模式为 <b>vpc</b> 。公网/VPC实例，部署模式为 <b>eip</b> 。
ZoneId	String	否	cn-hangzhou-k	实例部署的可用区ID。 <ul style="list-style-type: none"> <li>必须为vSwitch的可用区ID。</li> <li>格式可以为zoneX，也可以为RegionId-X。例如，可以设置为zonea或者cn-hangzhou-k。</li> </ul>
IsEipInner	Boolean	否	false	是否支持EIP。取值： <ul style="list-style-type: none"> <li><b>true</b>：公网/VPC实例</li> <li><b>false</b>：VPC实例</li> </ul> 该参数取值必须与实例类型一致。例如，实例类型为VPC实例，则此处必须设置为 <b>false</b> 。

名称	类型	是否必选	示例值	描述
IsSetUserAndPassword	Boolean	否	false	是否设置新的用户名和密码。取值： <ul style="list-style-type: none"> <li><b>true</b>：设置新的用户名和密码。</li> <li><b>false</b>：不设置新的用户名和密码。</li> </ul> 仅支持公网/VPC实例。
Username	String	否	username	用户名。 仅支持公网/VPC实例。
Password	String	否	password	用户密码。 仅支持公网/VPC实例。
Name	String	否	newInstanceName	实例名称。 <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 5px;"> <p><span style="color: #00aaff;">?</span> <b>说明</b> 如果传递此参数，请确保此参数在同地域内唯一。</p> </div>
SecurityGroup	String	否	sg-bp13wfx7kz9pkow***	实例的安全组。 不填写时，消息队列Kafka版会自动为您的实例配置安全组。如需填写，您需要先为实例创建安全组。具体操作，请参见 <a href="#">创建安全组</a> 。
ServiceVersion	String	否	0.10.2	部署的消息队列Kafka版的版本，可选值为0.10.2或2.2.0。

名称	类型	是否必选	示例值	描述
Config	String	否	{"kafka.log.retention.hours":"33"}	<p>部署的消息队列Kafka版的初始配置。配置信息必须是一个合法的JSON字符串。</p> <p>不填写时，该参数默认为空。</p> <p><b>Config</b>目前支持的参数配置如下：</p> <ul style="list-style-type: none"> <li>• <b>enable.vpc_sasl_ssl</b>：是否开启VPC传输加密。取值说明如下： <ul style="list-style-type: none"> <li>◦ <b>true</b>：开启VPC传输加密。如果开启，则须同时开启ACL。</li> <li>◦ <b>false</b>：默认值，不开启VPC传输加密。</li> </ul> </li> <li>• <b>enable.acl</b>：是否开启ACL。取值说明如下： <ul style="list-style-type: none"> <li>◦ <b>true</b>：开启ACL。</li> <li>◦ <b>false</b>：默认值，不开启ACL。</li> </ul> </li> <li>• <b>kafka.log.retention.hours</b>：在磁盘容量充足的情况下，消息的最长保留时间。单位：小时。取值范围[24, 480]，默认值72。磁盘容量不足（即磁盘水位达到85%）时，将提前删除旧的消息，以确保服务可用性。</li> <li>• <b>kafka.message.max.bytes</b>：消息队列Kafka版能收发的消息的最大值，单位：字节。取值范围[1048576, 10485760]，默认值1048576。修改该配置前，请确认修改值是否匹配生产和消费客户端相应配置。</li> </ul>
KMSKeyId	String	否	0d24xxxx-da7b-4786-b981-9a164dxxxxxx	<p>同地域内的云盘加密的密钥ID。您可以在<a href="#">密钥管理服务控制台</a>查看密钥ID，也可以创建新的密钥。具体操作，请参见<a href="#">管理密钥</a>。</p> <p>传入此参数表示开启实例加密（开启后无法关闭），在调用该接口时，系统会检查是否创建服务关联角色AliyunServiceRoleForAlikafkaInstanceEncryption，若未创建，则会自动创建服务关联角色。更多信息，请参见<a href="#">服务关联角色</a>。</p>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。



名称	类型	示例值	描述
Message	String	operation success.	返回信息。
RequestId	String	ABA4A7FD-E10F-45C7-9774-A5236015A***	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=StartInstance
&InstanceId=alikafka_post-cn-v0h1fgs2****
&RegionId=cn-hangzhou
&VpcId=vpc-bp1r4eg3yrxmygv****
&VSwitchId=vsw-bp1j3sg5979fstnpl****
&DeployModule=vpc
&ZoneId=cn-hangzhou-k
&IsEipInner=false
&IsSetUserAndPassword=false
&Username=username
&Password=password
&Name=newInstanceName
&SecurityGroup=sg-bp13wfx7kz9pkow***
&ServiceVersion= 0.10.2
&Config={"kafka.log.retention.hours":"33"}
&KMSKeyId=0d24xxxx-da7b-4786-b981-9a164dxxxxxx
&公共请求参数
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<StartInstanceResponse>
  <Code>200</Code>
  <Message>operation success.</Message>
  <RequestId>ABA4A7FD-E10F-45C7-9774-A5236015A***</RequestId>
  <Success>>true</Success>
</StartInstanceResponse>
```

#### JSON 格式

```

HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : 200,
  "Message" : "operation success.",
  "RequestId" : "ABA4A7FD-E10F-45C7-9774-A5236015A****",
  "Success" : true
}

```

## 错误码

访问[错误中心](#)查看更多错误码。

# 10.4. UpgradePostPayOrder

调用UpgradePostPayOrder升配后付费实例。

请确保在使用该接口前，已充分了解后付费实例的收费方式和价格。更多信息，请参见[计费说明](#)。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpgradePostPayOrder	要执行的操作。取值： <b>UpgradePostPayOrder</b> 。
DiskSize	Integer	是	1500	磁盘容量。 <ul style="list-style-type: none"> <li>填写的磁盘容量必须大于等于实例当前的磁盘容量。</li> <li>取值范围，请参见<a href="#">计费说明</a>。</li> </ul>
InstanceId	String	是	alibabacloud_post-cn-mp919o4v****	实例ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。
TopicQuota	Integer	是	80	Topic规格。 <ul style="list-style-type: none"> <li>填写的Topic规格必须大于或等于实例当前的Topic规格。</li> <li>流量规格不同，默认值不同。超过默认值，需额外收费。</li> <li>取值范围，请参见<a href="#">计费说明</a>。</li> </ul>

名称	类型	是否必选	示例值	描述
IoMax	Integer	否	90	<p>流量峰值（不推荐）。</p> <ul style="list-style-type: none"> <li>填写的流量峰值必须大于或等于实例当前的流量峰值。</li> <li>流量峰值和流量规格必须选填一个。同时填写时，以流量规格为准。建议您只填写流量规格。</li> <li>取值范围，请参见<a href="#">计费说明</a>。</li> </ul>
SpecType	String	否	professional	<p>规格类型。取值：</p> <ul style="list-style-type: none"> <li><b>normal</b>：普通版（高写版）</li> <li><b>professional</b>：专业版（高写版）</li> <li><b>professionalForHighRead</b>：专业版（高读版）</li> </ul> <p>不支持将实例从专业版降级到标准版。以上规格类型的说明请参见<a href="#">计费说明</a>。</p>
EipModel	Boolean	否	true	<p>是否需要公网。取值如下：</p> <ul style="list-style-type: none"> <li>true：需要公网。</li> <li>false：不需要公网。</li> </ul>
EipMax	Integer	否	300	<p>公网流量。</p> <ul style="list-style-type: none"> <li>填写的公网流量必须大于或等于实例当前的公网流量。</li> <li>取值范围，请参见<a href="#">计费说明</a>。</li> </ul> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>EipModel取值为true时，EipMax取值需大于0。</li> <li>EipModel取值为false时，EipMax取值需为0。</li> </ul> </div>
IoMaxSpec	String	否	alikafka.hw.6xlarge	<p>流量规格（推荐）。</p> <ul style="list-style-type: none"> <li>填写的流量规格必须大于或等于实例当前的流量规格。</li> <li>流量峰值和流量规格必须选填一个。同时填写时，以流量规格为准。建议您只填写流量规格。</li> <li>取值范围，请参见<a href="#">计费说明</a>。</li> </ul>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	ABA4A7FD-E10F-45C7-9774-A5236015A***	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=UpgradePostPayOrder
&InstanceId=alikafka_post-cn-m7r20thf****
&TopicQuota=80
&DiskSize=1500
&RegionId=cn-hangzhou
&IoMax=90
&SpecType=professional
&EipModel=true
&EipMax=300
&IoMaxSpec=alikafka.hw.6xlarge
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<UpgradePostPayOrderResponse>
  <Message>operation success.</Message>
  <RequestId>ABA4A7FD-E10F-45C7-9774-A5236015A***</RequestId>
  <Success>true</Success>
  <Code>200</Code>
</UpgradePostPayOrderResponse>
```

#### JSON 格式

```
{
  "RequestId": "ABA4A7FD-E10F-45C7-9774-A5236015A***",
  "Message": "operation success.",
  "Code": 200,
  "Success": true
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 10.5. UpgradePrePayOrder

调用UpgradePrePayOrder升配预付费实例。

请确保在使用该接口前，已充分了解预付费实例的收费方式和价格。详情请参见[计费说明](#)。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpgradePrePayOrder	要执行的操作。取值： <b>UpgradePrePayOrder</b> 。
DiskSize	Integer	是	900	磁盘容量。 <ul style="list-style-type: none"> <li>填写的磁盘容量必须大于等于实例当前的磁盘容量。</li> <li>取值范围请参见<a href="#">计费说明</a>。</li> </ul>
InstanceId	String	是	alikafka_post-cn-mp919o4v****	实例ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。
TopicQuota	Integer	是	50	Topic规格。 <ul style="list-style-type: none"> <li>填写的Topic规格必须大于或等于实例当前的Topic规格。</li> <li>流量规格不同，默认值不同。超过默认值，需额外收费。</li> <li>取值范围请参见<a href="#">计费说明</a>。</li> </ul>
IoMax	Integer	否	40	流量峰值（不推荐）。 <ul style="list-style-type: none"> <li>填写的流量峰值必须大于或等于实例当前的流量峰值。</li> <li>流量峰值和流量规格必须选填一个。同时填写时，以流量规格为准。建议您只填写流量规格。</li> <li>取值范围请参见<a href="#">计费说明</a>。</li> </ul>

名称	类型	是否必选	示例值	描述
SpecType	String	否	professional	规格类型。取值： <ul style="list-style-type: none"> <li>• <b>normal</b>：普通版（高写版）</li> <li>• <b>professional</b>：专业版（高写版）</li> <li>• <b>professionalForHighRead</b>：专业版（高读版）</li> </ul> 不支持将实例从专业版降级到标准版。以上规格类型的说明请参见 <a href="#">计费说明</a> 。
EipModel	Boolean	否	true	是否需要公网。取值如下： <ul style="list-style-type: none"> <li>• true：需要公网</li> <li>• false：不需要公网</li> </ul>
EipMax	Integer	否	3	公网流量。 <ul style="list-style-type: none"> <li>• 填写的公网流量必须大于或等于实例当前的公网流量。</li> <li>• 取值范围请参见<a href="#">计费说明</a>。</li> </ul> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>• EipModel取值为true时，EipMax取值需大于0。</li> <li>• EipModel取值为false时，EipMax取值需为0。</li> </ul> </div>
IoMaxSpec	String	否	alikafka.hw.2xlarge	流量规格（推荐）。 <ul style="list-style-type: none"> <li>• 填写的流量规格必须大于或等于实例当前的流量规格。</li> <li>• 流量峰值和流量规格必须选填一个。同时填写时，以流量规格为准。建议您只填写流量规格。</li> <li>• 取值范围请参见<a href="#">计费说明</a>。</li> </ul>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。

名称	类型	示例值	描述
RequestId	String	ABA4A7FD-E10F-45C7-9774-A5236015***	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=UpgradePrePayOrder
&RegionId=cn-hangzhou
&TopicQuota=50
&DiskSize=900
&InstanceId=alikafka_post-cn-mp919o4v****
&IoMax=40
&SpecType=professional
&EipModel=true
&EipMax=300
&IoMaxSpec=alikafka.hw.2xlarge
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<UpgradePrePayOrderResponse>
  <RequestId>ABA4A7FD-E10F-45C7-9774-A5236015***</RequestId>
  <Message>operation success.</Message>
  <Success>true</Success>
  <Code>200</Code>
</UpgradePrePayOrderResponse>
```

#### JSON 格式

```
{
  "RequestId": "ABA4A7FD-E10F-45C7-9774-A5236015***",
  "Message": "operation success.",
  "Code": "200",
  "Success": "true"
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 10.6. ConvertPostPayOrder

调用ConvertPostPayOrder将后付费实例转为预付费实例。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ConvertPostPayOrder	系统规定参数。取值： <b>ConvertPostPayOrder</b> 。
RegionId	String	是	cn-hangzhou	实例的地域ID。
InstanceId	String	是	alikafka_post-cn-v0h1fgs2****	实例的ID。
Duration	Integer	否	1	购买时长。单位为月。取值： <ul style="list-style-type: none"> <li>• 1~12</li> <li>• 24</li> <li>• 36</li> </ul>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	06084011-E093-46F3-A51F-4B19A8AD****	请求的ID。
Success	Boolean	true	调用是否成功。
OrderId	String	20497346575****	订单的ID。

## 示例

请求示例



```

http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=ConvertPostPayOrder
&RegionId=cn-hangzhou
&InstanceId=alikafka_post-cn-v0h1fgs2****
&Duration=1
&公共请求参数

```

### 正常返回示例

#### XML 格式

```

HTTP/1.1 200 OK
Content-Type:application/xml
<ConvertPostPayOrderResponse>
  <Code>200</Code>
  <Message>operation success.</Message>
  <RequestId>06084011-E093-46F3-A51F-4B19A8AD****</RequestId>
  <Success>>true</Success>
  <OrderId>20497346575****</OrderId>
</ConvertPostPayOrderResponse>

```

#### JSON 格式

```

HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : 200,
  "Message" : "operation success.",
  "RequestId" : "06084011-E093-46F3-A51F-4B19A8AD****",
  "Success" : true,
  "OrderId" : "20497346575****"
}

```

### 错误码

访问[错误中心](#)查看更多错误码。

## 10.7. ModifyInstanceName

调用ModifyInstanceName修改实例的名称。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
----	----	------	-----	----

名称	类型	是否必选	示例值	描述
Action	String	是	ModifyInstanceName	系统规定参数。取值： <b>ModifyInstanceName</b> 。
InstanceId	String	是	alikafka_post-cn-v0h1fgs2****	实例的ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。
InstanceName	String	是	dev-test	实例名称。取值： <ul style="list-style-type: none"> <li>只能包含字母，数字，下划线（_）和短划线（-）。</li> <li>长度限制在3~64字符，多于64字符将被自动截取。</li> </ul>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	06084011-E093-46F3-A51F-4B19A8AD****	请求的ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=ModifyInstanceName
&InstanceId=alikafka_post-cn-v0h1fgs2****
&RegionId=cn-hangzhou
&InstanceName=dev-test
&公共请求参数
```

### 正常返回示例

XML 格式

```

HTTP/1.1 200 OK
Content-Type:application/xml
<ModifyInstanceNameResponse>
  <Code>200</Code>
  <Message>operation success.</Message>
  <RequestId>06084011-E093-46F3-A51F-4B19A8AD****</RequestId>
  <Success>true</Success>
</ModifyInstanceNameResponse>

```

#### JSON 格式

```

HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : 200,
  "Message" : "operation success.",
  "RequestId" : "06084011-E093-46F3-A51F-4B19A8AD****",
  "Success" : true
}

```

## 错误码

访问[错误中心](#)查看更多错误码。

# 10.8. ReleaseInstance

调用ReleaseInstance接口释放后付费实例。

预付费实例不支持使用该接口释放。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ReleaseInstance	系统规定参数。取值： <b>ReleaseInstance</b> 。
InstanceId	String	是	alikafka_post-cn-mp919o4v****	实例ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。

名称	类型	是否必选	示例值	描述
ForceDeleteInstance	Boolean	否	false	是否立即释放实例的物理资源。取值： <ul style="list-style-type: none"> <li><b>true</b>：立即释放实例的物理资源。</li> <li><b>false</b>：实例物理资源会保留一段时间才释放。</li> </ul>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	ABA4A7FD-E10F-45C7-9774-A5236015A***	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=ReleaseInstance
&InstanceId=alikafka_post-cn-mp919o4v****
&RegionId=cn-hangzhou
&ForceDeleteInstance=false
&公共请求参数
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<ReleaseInstanceResponse>
  <Code>200</Code>
  <Message>operation success.</Message>
  <RequestId>ABA4A7FD-E10F-45C7-9774-A5236015A***</RequestId>
  <Success>>true</Success>
</ReleaseInstanceResponse>
```

#### JSON 格式

```

HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : 200,
  "Message" : "operation success.",
  "RequestId" : "ABA4A7FD-E10F-45C7-9774-A5236015A****",
  "Success" : true
}

```

## 错误码

访问[错误中心](#)查看更多错误码。

# 10.9. GetInstanceList

调用GetInstanceList查询指定地域的实例信息。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetInstanceList	系统规定参数。取值： <b>GetInstanceList</b> 。
RegionId	String	是	cn-hangzhou	实例的地域ID。
OrderId	String	否	20816072673****	订单ID。您可以在阿里云用户中心的 <a href="#">订单管理</a> 中获取。
InstanceId.N	String	否	alikafka_post-cn-mp91gnw0p***	实例ID。
Tag.N.Key	String	否	test	资源的标签键。 <ul style="list-style-type: none"> <li>N为1~20。</li> <li>如果为空，则匹配所有标签键。</li> <li>最多支持128个字符，不能以aliyun和acs:开头，不能包含http://或者https://。</li> </ul>

名称	类型	是否必选	示例值	描述
Tag.N.Value	String	否	test	资源的标签值。 <ul style="list-style-type: none"> <li>N为1~20。</li> <li>标签键为空时，必须为空。如果为空，匹配所有标签值。</li> <li>最多支持128个字符，不能以alibaba和acs:开头，不能包含http://或者https://。</li> </ul>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	4B6D821D-7F67-4CAA-9E13-A5A997C35***	请求ID。
Success	Boolean	true	调用是否成功。
InstanceList	Array of InstanceVO		实例详情。
InstanceVO			
VpcId	String	vpc-bp1ojac7bv448nifjl** *	VPC ID。
SpecType	String	professional	实例规格。取值： <ul style="list-style-type: none"> <li><b>professional</b>：专业版（高写版）</li> <li><b>professionalForHighRead</b>：专业版（高读版）</li> <li><b>normal</b>：标准版</li> </ul>
DeployType	Integer	5	部署类型。取值： <ul style="list-style-type: none"> <li><b>4</b>：公网/VPC实例</li> <li><b>5</b>：VPC实例</li> </ul>
CreateTime	Long	1577961819000	创建时间。

名称	类型	示例值	描述
DiskSize	Integer	3600	磁盘容量。
DiskType	Integer	1	磁盘类型。取值： <ul style="list-style-type: none"> <li>0：高效云盘</li> <li>1：SSD</li> </ul>
SecurityGroup	String	sg-bp13wfx7kz9pkowc***	实例的安全组。 <ul style="list-style-type: none"> <li>如果实例是通过消息队列Kafka版控制台或调用StartInstance（调用时没有配置安全组）部署的，则返回值为空。</li> <li>如果实例是通过调用StartInstance（调用时配置了安全组）部署的，则返回值为配置的安全组。</li> </ul>
SslEndPoint	String	192.0.2.XX:9093,198.51.100.XX:9093,203.0.113.XX:9093	SSL接入点的IP接入点。消息队列Kafka版实例支持域名接入点和IP接入点。 <ul style="list-style-type: none"> <li>域名接入点：由消息队列Kafka版实例域名和端口号拼接而成，格式为 {实例域名}:{端口号}。</li> <li>IP接入点：由Broker的IP地址和端口号拼接而成，格式为 {Broker的IP地址}:{端口号}。</li> </ul>
Instanceld	String	alikafka_pre-cn-mp919o4v****	实例ID。
AllConfig	String	{"enable.vpc_sasl_ssl":"false","kafka.log.retention.hours":"66","enable.acl":"false","kafka.message.max.bytes":"6291456"}	部署的消息队列Kafka版的当前配置。
ServiceStatus	Integer	5	实例状态。取值： <ul style="list-style-type: none"> <li>0：待部署</li> <li>1：部署中</li> <li>5：服务中</li> <li>15：已过期</li> </ul>
EipMax	Integer	20	公网流量峰值。
RegionId	String	cn-hangzhou	实例的地域ID。

名称	类型	示例值	描述
MsgRetain	Integer	72	消息保留时长。
VSwitchId	String	vsw-bp1fvuw0ljd7vzmo3d***	vSwitch ID。
ExpiredTime	Long	1893581018000	到期时间。
TopicNumLimit	Integer	180	Topic最大数量。
ZoneId	String	zonei	可用区ID。
IoMax	Integer	20	流量峰值。
PaidType	Integer	1	付费类型。取值： <ul style="list-style-type: none"> <li>0：预付费</li> <li>1：后付费</li> </ul>
Name	String	alikafka_post-cn-mp91gnw0p***	实例名称。
EndPoint	String	192.168.0.XX:9092,192.168.0.XX:9092,192.168.0.XX:9092	默认接入点的IP接入点。消息队列Kafka版实例支持域名接入点和IP接入点。 <ul style="list-style-type: none"> <li>域名接入点：由消息队列Kafka版实例域名和端口号拼接而成，格式为 {实例域名}:{端口号}。</li> <li>IP接入点：由Broker的IP地址和端口号拼接而成，格式为 {Broker的IP地址}:{端口号}。</li> </ul>
Tags	Array of TagVO		标签。
TagVO			
Key	String	test	标签键。
Value	String	test	标签值。
UpgradeServiceDetailInfo	Object		实例的升级信息。



名称	类型	示例值	描述
Current2OpenSourceVersion	String	2.2.0	实例的开源版本。
DomainEndpoint	String	alikafka-pre-cn-zv*****-1-vpc.alikafka.aliyuncs.com:9092,alikafka-pre-cn-zv*****-2-vpc.alikafka.aliyuncs.com:9092,alikafka-pre-cn-zv*****-3-vpc.alikafka.aliyuncs.com:9092	<p>默认接入点的域名接入点。消息队列Kafka版实例支持域名接入点和IP接入点。</p> <ul style="list-style-type: none"> <li>域名接入点：由消息队列Kafka版实例域名和端口号拼接而成，格式为 <code>{实例域名}:{端口号}</code>。</li> <li>IP接入点：由Broker的IP地址和端口号拼接而成，格式为 <code>{Broker的IP地址}:{端口号}</code>。</li> </ul>
SslDomainEndpoint	String	alikafka-pre-cn-zv*****-1.alikafka.aliyuncs.com:9093,alikafka-pre-cn-zv*****-2.alikafka.aliyuncs.com:9093,alikafka-pre-cn-zv*****-3.alikafka.aliyuncs.com:9093	<p>SSL接入点的域名接入点。消息队列Kafka版实例支持域名接入点和IP接入点。</p> <ul style="list-style-type: none"> <li>域名接入点：由消息队列Kafka版实例域名和端口号拼接而成，格式为 <code>{实例域名}:{端口号}</code>。</li> <li>IP接入点：由Broker的IP地址和端口号拼接而成，格式为 <code>{Broker的IP地址}:{端口号}</code>。</li> </ul>
SaslDomainEndpoint	String	alikafka-pre-cn-zv*****-1-vpc.alikafka.aliyuncs.com:9094,alikafka-pre-cn-zv*****-2-vpc.alikafka.aliyuncs.com:9094,alikafka-pre-cn-zv*****-3-vpc.alikafka.aliyuncs.com:9094	<p>SASL接入点的域名接入点。消息队列Kafka版实例支持域名接入点和IP接入点。</p> <ul style="list-style-type: none"> <li>域名接入点：由消息队列Kafka版实例域名和端口号拼接而成，格式为 <code>{实例域名}:{端口号}</code>。</li> <li>IP接入点：由Broker的IP地址和端口号拼接而成，格式为 <code>{Broker的IP地址}:{端口号}</code>。</li> </ul>

## AllConfig参数说明

名称	类型	取值范围	示例值	描述
enable.vpc_sasl_ssl	Boolean	<ul style="list-style-type: none"> <li>true</li> <li>false</li> </ul>	false	是否开启VPC传输加密，如果要开启，必须同时开启ACL。

名称	类型	取值范围	示例值	描述
kafka.log.retention.hours	Integer	24 ~ 480	66	消息保留时长（小时）。
enable.acl	Boolean	<ul style="list-style-type: none"><li>true</li><li>false</li></ul>	false	是否开启ACL。
kafka.message.max.bytes	Integer	1048576 ~ 10485760	6291456	最大消息大小（字节）。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=GetInstanceList
&RegionId=cn-hangzhou
&OrderId=20816072673****
&InstanceId=["alikafka_post-cn-mp91gnw0p****"]
&Tag=[{"Key":"test","Value":"test"}]
&公共请求参数
```

### 正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<GetInstanceListResponse>
  <Code>200</Code>
  <Message>operation success.</Message>
  <RequestId>4B6D821D-7F67-4CAA-9E13-A5A997C35***</RequestId>
  <Success>>true</Success>
  <InstanceList>
    <VpcId>vpc-bp1ojac7bv448nifj1***</VpcId>
    <SpecType>professional</SpecType>
    <DeployType>5</DeployType>
    <CreateTime>1577961819000</CreateTime>
    <DiskSize>3600</DiskSize>
    <DiskType>1</DiskType>
    <SecurityGroup>sg-bp13wfx7kz9pkowc***</SecurityGroup>
    <SslEndPoint>192.0.2.XX:9093,198.51.100.XX:9093,203.0.113.XX:9093</SslEndPoint>
    <InstanceId>alikafka_pre-cn-mp919o4v***</InstanceId>
    <AllConfig>{"enable.vpc_sasl_ssl":"false","kafka.log.retention.hours":"66",
  \ "enable.acl":"false","kafka.message.max.bytes":"6291456"}</AllConfig>
    <ServiceStatus>5</ServiceStatus>
    <EipMax>20</EipMax>
    <RegionId>cn-hangzhou</RegionId>
    <MsgRetain>72</MsgRetain>
    <VSwitchId>vsw-bp1fvuw0lj7vzmo3d***</VSwitchId>
    <ExpiredTime>1893581018000</ExpiredTime>
    <TopicNumLimit>180</TopicNumLimit>
    <ZoneId>zonei</ZoneId>
    <IoMax>20</IoMax>
    <PaidType>1</PaidType>
    <Name>alikafka_post-cn-mp91gnw0p***</Name>
    <EndPoint>192.168.0.XX:9092,192.168.0.XX:9092,192.168.0.XX:9092</EndPoint>
    <Tags>
      <Key>test</Key>
      <Value>test</Value>
    </Tags>
    <UpgradeServiceDetailInfo>
      <Current2OpenSourceVersion>2.2.0</Current2OpenSourceVersion>
    </UpgradeServiceDetailInfo>
    <DomainEndpoint>alikafka-pre-cn-zv*****-1-vpc.alikafka.aliyuncs.com:9092,alika
  fka-pre-cn-zv*****-2-vpc.alikafka.aliyuncs.com:9092,alikafka-pre-cn-zv*****-3-vpc
  .alikafka.aliyuncs.com:9092</DomainEndpoint>
    <SslDomainEndpoint>alikafka-pre-cn-zv*****-1.alikafka.aliyuncs.com:9093,alika
  fka-pre-cn-zv*****-2.alikafka.aliyuncs.com:9093,alikafka-pre-cn-zv*****-3.alikafka
  .aliyuncs.com:9093</SslDomainEndpoint>
    <SaslDomainEndpoint>alikafka-pre-cn-zv*****-1-vpc.alikafka.aliyuncs.com:9094,a
  likafka-pre-cn-zv*****-2-vpc.alikafka.aliyuncs.com:9094,alikafka-pre-cn-zv*****-3
  -vpc.alikafka.aliyuncs.com:9094</SaslDomainEndpoint>
  </InstanceList>
</GetInstanceListResponse>
```

JSON 格式

```

HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : 200,
  "Message" : "operation success.",
  "RequestId" : "4B6D821D-7F67-4CAA-9E13-A5A997C35***",
  "Success" : true,
  "InstanceList" : [ {
    "VpcId" : "vpc-bplojac7bv448nifj1***",
    "SpecType" : "professional",
    "DeployType" : 5,
    "CreateTime" : 1577961819000,
    "DiskSize" : 3600,
    "DiskType" : 1,
    "SecurityGroup" : "sg-bp13wfx7kz9pkowc***",
    "SslEndPoint" : "192.0.2.XX:9093,198.51.100.XX:9093,203.0.113.XX:9093",
    "InstanceId" : "alikafka_pre-cn-mp919o4v***",
    "AllConfig" : "{\\"enable.vpc_sasl_ssl\\":\\"false\\",\\"kafka.log.retention.hours\\":\\"66\\",\\"enable.ac1\\":\\"false\\",\\"kafka.message.max.bytes\\":\\"6291456\\"}",
    "ServiceStatus" : 5,
    "EipMax" : 20,
    "RegionId" : "cn-hangzhou",
    "MsgRetain" : 72,
    "VSwitchId" : "vsw-bp1fvuw0ljd7vzmo3d***",
    "ExpiredTime" : 1893581018000,
    "TopicNumLimit" : 180,
    "ZoneId" : "zonei",
    "IoMax" : 20,
    "PaidType" : 1,
    "Name" : "alikafka_post-cn-mp91gnw0p***",
    "EndPoint" : "192.168.0.XX:9092,192.168.0.XX:9092,192.168.0.XX:9092",
    "Tags" : [ {
      "Key" : "test",
      "Value" : "test"
    } ],
    "UpgradeServiceDetailInfo" : {
      "Current2OpenSourceVersion" : "2.2.0"
    },
    "DomainEndpoint" : "alikafka-pre-cn-zv*****-1-vpc.alikafka.aliyuncs.com:9092,alikaf
    ka-pre-cn-zv*****-2-vpc.alikafka.aliyuncs.com:9092,alikafka-pre-cn-zv*****-3-vpc
    .alikafka.aliyuncs.com:9092",
    "SslDomainEndpoint" : "alikafka-pre-cn-zv*****-1.alikafka.aliyuncs.com:9093,alika
    ka-pre-cn-zv*****-2.alikafka.aliyuncs.com:9093,alikafka-pre-cn-zv*****-3.alikafka
    .aliyuncs.com:9093",
    "SaslDomainEndpoint" : "alikafka-pre-cn-zv*****-1-vpc.alikafka.aliyuncs.com:9094,a
    likafka-pre-cn-zv*****-2-vpc.alikafka.aliyuncs.com:9094,alikafka-pre-cn-zv*****-3
    -vpc.alikafka.aliyuncs.com:9094"
  } ]
}

```

## 错误码

访问[错误中心](#)查看更多错误码。

## 10.10. GetAllInstanceIdList

调用GetAllInstanceIdList查询当前用户下所有实例ID集合。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetAllInstanceIdList	系统规定参数。取值： <b>GetAllInstanceIdList</b> 。
RegionId	String	是	cn-hangzhou	实例的地域ID。保留字段。

### 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
InstanceIds	Map		当前用户下所有地域的实例ID集合。
RequestId	String	ABA4A7FD-E10F-45C7-9774-A5236015****	请求ID。
Success	Boolean	true	调用是否成功。

### 示例

#### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=GetAllInstanceIdList
&RegionId=cn-hangzhou
&公共请求参数
```

#### 正常返回示例

XML 格式

```

HTTP/1.1 200 OK
Content-Type:application/xml
<GetAllInstanceIdListResponse>
  <Message>operation success.</Message>
  <RequestId>ABA4A7FD-E10F-45C7-9774-A5236015****</RequestId>
  <InstanceIds>
    <cn-shenzhen>alikaafka_post-cn-7pp2btvo****</cn-shenzhen>
    <us-west-1>alikaafka_pre-cn-i7m2l1xid****</us-west-1>
    <cn-hangzhou>alikaafka_pre-cn-i7m2hflj****</cn-hangzhou>
    <cn-hangzhou>alikaafka_pre-cn-zvp2hsje****</cn-hangzhou>
    <cn-hangzhou>alikaafka_pre-cn-zvp2kvc9****</cn-hangzhou>
  </InstanceIds>
  <Code>200</Code>
  <Success>>true</Success>
</GetAllInstanceIdListResponse>

```

JSON 格式

```

HTTP/1.1 200 OK
Content-Type:application/json
{
  "Message" : "operation success.",
  "RequestId" : "ABA4A7FD-E10F-45C7-9774-A5236015****",
  "InstanceIds" : {
    "cn-shenzhen" : [ "alikaafka_post-cn-7pp2btvo****" ],
    "us-west-1" : [ "alikaafka_pre-cn-i7m2l1xid****" ],
    "cn-hangzhou" : [ "alikaafka_pre-cn-i7m2hflj****", "alikaafka_pre-cn-zvp2hsje****", "alikaafka_pre-cn-zvp2kvc9****" ]
  },
  "Code" : 200,
  "Success" : true
}

```

### 错误码

访问[错误中心](#)查看更多错误码。

## 10.11. DeleteInstance

调用DeleteInstance删除实例。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
----	----	------	-----	----

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteInstance	要执行的操作。取值： <b>DeleteInstance</b> 。
InstanceId	String	是	alikafka_post-cn-mp919o4v****	实例ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	ABA4A7FD-E10F-45C7-9774-A5236015****	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DeleteInstance
&InstanceId=alikafka_post-cn-mp919o4v****
&RegionId=cn-hangzhou
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DeleteInstanceResponse>
  <Message>operation success.</Message>
  <RequestId>ABA4A7FD-E10F-45C7-9774-A5236015****</RequestId>
  <Success>>true</Success>
  <Code>200</Code>
</DeleteInstanceResponse>
```

#### JSON 格式

```

{
  "RequestId": "ABA4A7FD-E10F-45C7-9774-A5236015****",
  "Message": "operation success.",
  "Code": "200",
  "Success": "true"
}

```

### 错误码

访问[错误中心](#)查看更多错误码。

## 10.12. GetAllowedIpList

调用GetAllowedIpList获取IP白名单。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetAllowedIpList	要执行的操作。取值： <b>GetAllowedIpList</b> 。
InstanceId	String	是	alikafka_post-cn-mp91inkw****	实例ID。
RegionId	String	是	cn-hangzhou	地域ID。

### 返回数据

名称	类型	示例值	描述
AllowedList	Struct		白名单。
DeployType	Integer	4	部署类型。取值： <ul style="list-style-type: none"> <li>4: 公网/VPC</li> <li>5: VPC</li> </ul> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p><span style="color: #00aaff;">?</span> <b>说明</b> 普通用户无需关心该字段，集成商客户可以了解一下。</p> </div>



名称	类型	示例值	描述
InternetList	Array of IPlistVO		公网白名单。
AllowedIpList	List	0.0.0.0/0	IP白名单。
PortRange	String	9093/9093	端口范围。取值： <b>9093/9093</b> 。
Vpclist	Array of IPlistVO		VPC白名单。
AllowedIpList	List	192.XXX.X.X/XX	IP白名单。
PortRange	String	9092/9092	端口范围。取值： <b>9092/9092</b> 。
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	A421CCD7-5BC5-4B32-8DD8-64668A8FCB56	请求的ID。
Success	Boolean	true	请求是否成功。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=GetAllowedIpList
&RegionId=cn-hangzhou
&InstanceId=alikaafka_post-cn-mp9linkw****
&<公共请求参数>
```

### 正常返回示例

XML 格式

```
<GetAllowedIpListResponse>
  <RequestId>A421CCD7-5BC5-4B32-8DD8-64668A*****</RequestId>
  <Message>operation success.</Message>
  <AllowedList>
    <DeployType>4</DeployType>
    <InternetList>
      <PortRange>9093/9093</PortRange>
      <AllowedIpList>0.0.0.0/0</AllowedIpList>
    </InternetList>
    <VpcList>
      <PortRange>9094/9094</PortRange>
      <AllowedIpList>0.0.0.0/0</AllowedIpList>
    </VpcList>
    <VpcList>
      <PortRange>9092/9092</PortRange>
      <AllowedIpList>0.0.0.0/0</AllowedIpList>
      <AllowedIpList>192.XXX.X.X/XX</AllowedIpList>
    </VpcList>
  </AllowedList>
  <Code>200</Code>
  <Success>>true</Success>
</GetAllowedIpListResponse>
```

JSON 格式

```
{
  "RequestId": "A421CCD7-5BC5-4B32-8DD8-64668A*****",
  "Message": "operation success.",
  "AllowedList": {
    "DeployType": 4,
    "InternetList": [
      {
        "PortRange": "9093/9093",
        "AllowedIpList": [
          "0.0.0.0/0"
        ]
      }
    ],
    "VpcList": [
      {
        "PortRange": "9094/9094",
        "AllowedIpList": [
          "0.0.0.0/0"
        ]
      },
      {
        "PortRange": "9092/9092",
        "AllowedIpList": [
          "0.0.0.0/0",
          "192.XXX.X.X/XX"
        ]
      }
    ]
  },
  "Code": 200,
  "Success": true
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 10.13. UpdateAllowedIp

调用UpdateAllowedIp变更IP白名单。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
----	----	------	-----	----

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateAllowedIp	要执行的操作。取值： <b>UpdateAllowedIp</b> 。
AllowedListIp	String	是	0.0.0.0/0	IP列表。可以是网段，例如： <b>192.168.0.0/16</b> 。 <ul style="list-style-type: none"> <li>当UpdateType为add时，可以填写多项，以英文逗号(,)分隔。</li> <li>当UpdateType为delete时，一次只能填写单项。</li> <li>删除需谨慎。</li> </ul>
AllowedListType	String	是	vpc	白名单的类型。取值： <ul style="list-style-type: none"> <li><b>vpc</b>：专有网络VPC。</li> <li><b>internet</b>：公网。</li> </ul>
InstanceId	String	是	alikafka_pre-cn-0pp1cng20***	实例ID。
PortRange	String	是	9092/9092	端口范围。取值： <ul style="list-style-type: none"> <li><b>9092/9092</b>：专有网络VPC。</li> <li><b>9093/9093</b>：公网。</li> </ul> 该参数需要与AllowedListType对应。
RegionId	String	是	cn-hangzhou	实例的地域ID。
UpdateType	String	是	add	变更类型。取值： <ul style="list-style-type: none"> <li><b>add</b>：增加。</li> <li><b>delete</b>：删除。</li> </ul>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。

名称	类型	示例值	描述
RequestId	String	17D425C2-4EA3-4AB8-928D-E10511ECF***	请求ID。
Success	Boolean	true	请求是否成功。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=UpdateAllowedIp
&RegionId=cn-hangzhou
&UpdateType=add
&PortRange=9092/9092
&AllowedListType=vpc
&AllowedListIp=0.0.0.0/0
&InstanceId=alikafka_pre-cn-0pp1cng20***
&<公共请求参数>
```

### 正常返回示例

XML 格式

```
<UpdateAllowedIpResponse>
  <RequestId>17D425C2-4EA3-4AB8-928D-E10511ECF***</RequestId>
  <Message>operation success.</Message>
  <Code>200</Code>
  <Success>>true</Success>
</UpdateAllowedIpResponse>
```

JSON 格式

```
{
  "RequestId": "17D425C2-4EA3-4AB8-928D-E10511ECF***",
  "Message": "operation success.",
  "Code": 200,
  "Success": true
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 10.14. UpdateInstanceConfig

调用UpdateInstanceConfig接口更新实例配置。

## 权限说明

RAM用户需要先获取授权，才能调用UpdateInstanceConfig接口。授权的详细信息，请参见[RAM权限策略](#)。

API	Action	Resource
UpdateInstanceConfig	alikafka: UpdateInstance	acs:alikafka:*:*:{instanceId}

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateInstanceConfig	系统规定参数。取值：UpdateInstanceConfig。
Config	String	是	{"kafka.log.retention.hours":"33"}	需要更新的消息队列Kafka版的配置。配置信息必须是一个合法的JSON字符串。
InstanceId	String	是	alikafka_post-cn-v0h1fgs2****	实例的ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。

## Config参数说明

名称	类型	取值范围	默认值	描述
enable.vpc_sasl_ssl	Boolean	true/false	false	是否开启VPC传输加密，如果要开启，必须同时开启ACL。
enable.acl	Boolean	true/false	false	是否开启ACL。
kafka.log.retention.hours	Integer	24~480	72	消息保留时长（小时）。
kafka.message.max.bytes	Integer	1048576~10485760	1048576	最大消息大小（字节）。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	4B6D821D-7F67-4CAA-9E13-A5A997C35***	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=UpdateInstanceConfig
&Config={"kafka.log.retention.hours":"33"}
&InstanceId=alikafka_post-cn-v0h1fgs2****
&RegionId=cn-hangzhou
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<UpdateInstanceConfigResponse>
  <Message>operation success.</Message>
  <RequestId>4B6D821D-7F67-4CAA-9E13-A5A997C35***</RequestId>
  <Code>200</Code>
  <Success>>true</Success>
</UpdateInstanceConfigResponse>
```

#### JSON 格式

```
{
  "Message": "operation success.",
  "RequestId": "4B6D821D-7F67-4CAA-9E13-A5A997C35***",
  "Code": 200,
  "Success": true
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 10.15. UpgradeInstanceVersion

调用UpgradeInstanceVersion接口升配实例版本。

## 权限说明

RAM用户需要先获取授权，才能调用UpgradeInstanceVersion接口。授权的详细信息，请参见[RAM权限策略](#)。

API	Action	Resource
UpgradeInstanceVersion	UpdateInstance	acs:alikafka:*:*:{instanceId}

## QPS限制

单用户请求频率限制为2 QPS。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpgradeInstanceVersion	系统规定参数。取值： <b>UpgradeInstanceVersion</b> 。
InstanceId	String	是	alikafka_post-cn-v0h1fgs2****	实例的ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。
TargetVersion	String	是	0.10.2	目标开源版本。可取值为： <ul style="list-style-type: none"> <li>0.10.2</li> <li>2.2.0</li> </ul> 如果填写值为同版本，会触发升级到最新小版本。

## TargetVersion参数示例说明

升级前开源版本	升级前小版本	目标开源版本	升级后开源版本	升级后小版本
0.10.2	非最新小版本	0.10.2	0.10.2	最新小版本
0.10.2	非最新小版本	2.2.0	2.2.0	最新小版本



升级前开源版本	升级前小版本	目标开源版本	升级后开源版本	升级后小版本
0.10.2	最新小版本	2.2.0	2.2.0	最新小版本
2.2.0	非最新小版本	2.2.0	2.2.0	最新小版本

升级小版本详细信息，请参见[升级实例版本](#)。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	ABA4A7FD-E10F-45C7-9774-A5236015***	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=UpgradeInstanceVersion
&InstanceId=alikafka_post-cn-v0h1fgs2****
&RegionId=cn-hangzhou
&TargetVersion=0.10.2
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<UpgradeInstanceVersionResponse>
  <Message>operation success.</Message>
  <RequestId>ABA4A7FD-E10F-45C7-9774-A5236015***</RequestId>
  <Code>200</Code>
  <Success>true</Success>
</UpgradeInstanceVersionResponse>
```

#### JSON 格式

```
{
  "Message": "operation success.",
  "RequestId": "ABA4A7FD-E10F-45C7-9774-A5236015***",
  "Code": 200,
  "Success": true
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 11.Topic

## 11.1. CreateTopic

调用CreateTopic创建Topic。



- 单用户请求频率限制为1 QPS。
- 每个实例下最多可创建的Topic数量与您所购买的实例规格相关。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateTopic	系统规定参数。取值： <b>CreateTopic</b> 。
Instanceid	String	是	alikafka_pre-cn-mp919o4v****	实例ID。
Topic	String	是	alikafka_topic_test	Topic的名称。 <ul style="list-style-type: none"> <li>• 只能包含字母、数字、下划线（_）和短划线（-）。</li> <li>• 长度限制为3~64字符，多于64个字符将被自动截取。</li> <li>• Topic名称一旦创建，将无法修改。</li> </ul>
Remark	String	是	alikafka_topic_test	Topic的备注。 <ul style="list-style-type: none"> <li>• 只能包含字母、数字、下划线（_）、短划线（-）。</li> <li>• 长度为3~64字符。</li> </ul>
Regionid	String	是	cn-hangzhou	Topic所在实例的地域ID。
CompactTopic	Boolean	否	false	Topic的存储引擎配置为Local存储时，会配置日志清理策略。取值： <ul style="list-style-type: none"> <li>• false: delete清理策略。</li> <li>• true: compact清理策略。</li> </ul>

名称	类型	是否必选	示例值	描述
PartitionNum	String	否	12	<p>Topic的分区数。</p> <ul style="list-style-type: none"> <li>分区数限制1~360。</li> <li>控制台会根据实例规格提供不同的配置建议，请根据控制台建议配置分区数，减少数据倾斜风险。</li> <li>特殊需求请提交工单。</li> </ul>
LocalTopic	Boolean	否	false	<p>Topic的存储引擎。取值：</p> <ul style="list-style-type: none"> <li>false：云存储。</li> <li>true：Local存储。</li> </ul>
ReplicationFactor	Long	否	3	<p>Topic的副本数。</p> <ul style="list-style-type: none"> <li>仅当LocalTopic值为true，才可以指定该参数。</li> <li>副本数限制1~3。</li> </ul> <p> 说明 副本数为1时，有数据丢失的风险，请谨慎设置。</p>
MinInsyncReplicas	Long	否	1	<p>最小ISR同步副本数。</p> <ul style="list-style-type: none"> <li>仅当LocalTopic值为true，才可以指定该参数。</li> <li>取值需小于Topic副本数。</li> <li>同步副本数限制1~3。</li> </ul>
Config	Map	否	{"replications": 3}	<p>补充配置。</p> <ul style="list-style-type: none"> <li>必须为JSON格式。</li> <li>目前仅支持Key为replications。表示Topic副本数，取值类型为Integer，取值限制1~3。</li> <li>仅当LocalTopic值为true，才可以指定该参数。</li> </ul> <p> 说明 如果该参数中指定了replications，指定的ReplicationFactor参数不再生效。</p>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success	返回信息。
RequestId	String	9C0F207C-77A6-43E5-991C-9D98510A****	请求的ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=CreateTopic
&InstanceId=alikafka_pre-cn-mp919o4v****
&RegionId=cn-hangzhou
&Remark=alikafka_topic_test
&Topic=alikafka_topic_test
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<CreateTopicResponse>
  <RequestId>9C0F207C-77A6-43E5-991C-9D98510A****</RequestId>
  <Message>operation success</Message>
  <Code>200</Code>
  <Success>>true</Success>
</CreateTopicResponse>
```

#### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "9C0F207C-77A6-43E5-991C-9D98510A****",
  "Message" : "operation success",
  "Code" : 200,
  "Success" : true
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

## 11.2. DeleteTopic

调用DeleteTopic删除Topic。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteTopic	要执行的操作。取值： <b>DeleteTopic</b> 。
InstanceId	String	是	alikafka_post-cn-v0h1fgs2****	实例ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。
Topic	String	是	test	Topic名称。

### 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	06084011-E093-46F3-A51F-4B19A8AD****	请求的ID。
Success	Boolean	true	调用是否成功。

### 示例

请求示例

```
http(s)://[Endpoint]/?Action=DeleteTopic
&InstanceId=alikafka_post-cn-v0h1fgs2****
&RegionId=cn-hangzhou
&Topic=test
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DeleteTopicResponse>
  <RequestId>06084011-E093-46F3-A51F-4B19A8AD****</RequestId>
  <Message>operation success.</Message>
  <Code>200</Code>
  <Success>>true</Success>
</DeleteTopicResponse>
```

#### JSON 格式

```
{
  "RequestId": "06084011-E093-46F3-A51F-4B19A8AD****",
  "Message": "operation success.",
  "Code": 200,
  "Success": true
}
```

### 错误码

访问[错误中心](#)查看更多错误码。

## 11.3. GetTopicList

调用GetTopicList获取Topic信息。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetTopicList	要执行的操作。取值： <b>GetTopicList</b> 。
InstanceId	String	是	alikafka_pre-cn-0pp1954n****	Topic所属实例的ID。
CurrentPage	String	否	1	当前页码。

名称	类型	是否必选	示例值	描述
PageSize	String	否	10	分页大小。
RegionId	String	否	cn-hangzhou	Topic所属实例的地域ID。
Topic	String	否	topic_name	获取指定Topic信息的Topic名称。

## 返回数据

名称	类型	示例值	描述
CurrentPage	Integer	1	当前页码。
RequestId	String	C0D3DC5B-5C37-47AD-9F22-1F5598809***	请求ID。
Success	Boolean	true	调用是否成功。
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
PageSize	Integer	10	页大小。
Total	Integer	1	Topic总数。
TopicList	Array of TopicVO		Topic详情。
TopicVO			
Status	Integer	0	服务状态。取值： 0：服务中。 删除Topic，则Topic无服务状态。
PartitionNum	Integer	6	Topic的分区数量。



名称	类型	示例值	描述
Remark	String	test	备注。取值： <ul style="list-style-type: none"> <li>只能包含字母、数字、下划线（_）和短划线（-）。</li> <li>长度为3~64字符。</li> </ul>
CreateTime	Long	1576563109000	创建时间。
Topic	String	topic_name	Topic名称。取值： <ul style="list-style-type: none"> <li>只能包含字母、数字、下划线（_）和短划线（-）。</li> <li>限制在3~64字符，长于64字符将被自动截取。</li> </ul>
StatusName	String	服务中	服务状态名称。取值： <b>服务中。</b> 删除Topic，则Topic无服务状态名称。
CompactTopic	Boolean	false	<b>LocalTopic</b> 配置为true时，会配置日志清理策略。取值： <ul style="list-style-type: none"> <li>false: delete清理策略。</li> <li>true: compact清理策略。</li> </ul>
InstanceId	String	alibaba_pre-cn-0pp1954n****	实例ID。
LocalTopic	Boolean	false	Topic的存储引擎。取值： <ul style="list-style-type: none"> <li>false: 云存储。</li> <li>true: Local存储。</li> </ul>
RegionId	String	cn-hangzhou	Topic所属实例的地域ID。
Tags	Array of TagVO		标签。
TagVO			
Key	String	Test	标签键。
Value	String	Test	标签值。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=GetTopicList
&InstanceId=alikafka_pre-cn-0pp1954n****
&CurrentPage=1
&PageSize=10
&RegionId=cn-hangzhou
&Topic=topic_name
&公共请求参数
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<GetTopicListResponse>
  <CurrentPage>1</CurrentPage>
  <RequestId>C0D3DC5B-5C37-47AD-9F22-1F5598809****</RequestId>
  <Success>true</Success>
  <Code>200</Code>
  <Message>operation success.</Message>
  <PageSize>10</PageSize>
  <Total>1</Total>
  <TopicList>
    <Status>0</Status>
    <PartitionNum>6</PartitionNum>
    <Remark>test</Remark>
    <CreateTime>1576563109000</CreateTime>
    <Topic>topic_name</Topic>
    <StatusName>服务中</StatusName>
    <CompactTopic>>false</CompactTopic>
    <InstanceId>alikafka_pre-cn-0pp1954n****</InstanceId>
    <LocalTopic>>false</LocalTopic>
    <RegionId>cn-hangzhou</RegionId>
    <Tags>
      <Key>Test</Key>
      <Value>Test</Value>
    </Tags>
  </TopicList>
</GetTopicListResponse>
```

#### JSON 格式

```

HTTP/1.1 200 OK
Content-Type:application/json
{
  "CurrentPage" : 1,
  "RequestId" : "C0D3DC5B-5C37-47AD-9F22-1F5598809****",
  "Success" : true,
  "Code" : 200,
  "Message" : "operation success.",
  "PageSize" : 10,
  "Total" : 1,
  "TopicList" : [ {
    "Status" : 0,
    "PartitionNum" : 6,
    "Remark" : "test",
    "CreateTime" : 1576563109000,
    "Topic" : "topic_name",
    "StatusName" : "服务中",
    "CompactTopic" : false,
    "InstanceId" : "alikaafka_pre-cn-0pp1954n****",
    "LocalTopic" : false,
    "RegionId" : "cn-hangzhou",
    "Tags" : [ {
      "Key" : "Test",
      "Value" : "Test"
    } ]
  } ]
} ]
}

```

## 错误码

访问[错误中心](#)查看更多错误码。

# 11.4. GetTopicStatus

调用GetTopicStatus获取Topic的消息收发状态。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetTopicStatus	要执行的操作。取值： <b>GetTopicStatus</b> 。
InstanceId	String	是	alikaafka_pre-cn-v0h15tjmo003	实例ID。

名称	类型	是否必选	示例值	描述
Topic	String	是	normal_topic_9d034262835916103455551be06cc2dc_6	Topic名称。
RegionId	String	否	cn-hangzhou	地域ID。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	状态码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	E475C7E2-8C35-46EF-BE7D-5D2A9F5D****	请求ID。
Success	Boolean	true	调用是否成功。
TopicStatus	Struct		Topic状态。
LastTimeStamp	Long	1566470063575	最后一条被消费的消息的产生时间。
OffsetTable	Array		偏移列表。
OffsetTable			
LastUpdateTimeStamp	Long	1566470063547	最后更新时间。
MaxOffset	Long	76	消息最大位点。
MinOffset	Long	0	消息最小位点。
Partition	Integer	0	分区ID。
Topic	String	testkafka	Topic名称。

名称	类型	示例值	描述
TotalCount	Long	423	消息总数。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=GetTopicStatus
&InstanceId=alikafka_pre-cn-v0h15tjmo003
&Topic=normal_topic_9d034262835916103455551be06cc2dc_6
&<公共请求参数>
```

### 正常返回示例

XML 格式

```
<GetTopicStatusResponse>
  <Message>operation success.</Message>
  <RequestId>D2FF4C1A-2A4A-4C24-BFCD-A2FF2DC1AAFE</RequestId>
  <TopicStatus>
    <TotalCount>0</TotalCount>
    <LastTimeStamp>0</LastTimeStamp>
    <OffsetTable>
      <OffsetTable>
        <Partition>0</Partition>
        <LastUpdateTimestamp>0</LastUpdateTimestamp>
        <Topic>demo</Topic>
        <MaxOffset>0</MaxOffset>
        <MinOffset>0</MinOffset>
      </OffsetTable>
      <OffsetTable>
        <Partition>1</Partition>
        <LastUpdateTimestamp>0</LastUpdateTimestamp>
        <Topic>demo</Topic>
        <MaxOffset>0</MaxOffset>
        <MinOffset>0</MinOffset>
      </OffsetTable>
      <OffsetTable>
        <Partition>2</Partition>
        <LastUpdateTimestamp>0</LastUpdateTimestamp>
        <Topic>demo</Topic>
        <MaxOffset>0</MaxOffset>
        <MinOffset>0</MinOffset>
      </OffsetTable>
      <OffsetTable>
        <Partition>3</Partition>
        <LastUpdateTimestamp>0</LastUpdateTimestamp>
        <Topic>demo</Topic>
        <MaxOffset>0</MaxOffset>
        <MinOffset>0</MinOffset>
      </OffsetTable>
    </OffsetTable>
  </TopicStatus>
</GetTopicStatusResponse>
```

```
<Partition>4</Partition>
<LastUpdateTimestamp>0</LastUpdateTimestamp>
<Topic>demo</Topic>
<MaxOffset>0</MaxOffset>
<MinOffset>0</MinOffset>
</OffsetTable>
<OffsetTable>
  <Partition>5</Partition>
  <LastUpdateTimestamp>0</LastUpdateTimestamp>
  <Topic>demo</Topic>
  <MaxOffset>0</MaxOffset>
  <MinOffset>0</MinOffset>
</OffsetTable>
<OffsetTable>
  <Partition>6</Partition>
  <LastUpdateTimestamp>0</LastUpdateTimestamp>
  <Topic>demo</Topic>
  <MaxOffset>0</MaxOffset>
  <MinOffset>0</MinOffset>
</OffsetTable>
<OffsetTable>
  <Partition>7</Partition>
  <LastUpdateTimestamp>0</LastUpdateTimestamp>
  <Topic>demo</Topic>
  <MaxOffset>0</MaxOffset>
  <MinOffset>0</MinOffset>
</OffsetTable>
<OffsetTable>
  <Partition>8</Partition>
  <LastUpdateTimestamp>0</LastUpdateTimestamp>
  <Topic>demo</Topic>
  <MaxOffset>0</MaxOffset>
  <MinOffset>0</MinOffset>
</OffsetTable>
<OffsetTable>
  <Partition>9</Partition>
  <LastUpdateTimestamp>0</LastUpdateTimestamp>
  <Topic>demo</Topic>
  <MaxOffset>0</MaxOffset>
  <MinOffset>0</MinOffset>
</OffsetTable>
<OffsetTable>
  <Partition>10</Partition>
  <LastUpdateTimestamp>0</LastUpdateTimestamp>
  <Topic>demo</Topic>
  <MaxOffset>0</MaxOffset>
  <MinOffset>0</MinOffset>
</OffsetTable>
<OffsetTable>
  <Partition>11</Partition>
  <LastUpdateTimestamp>0</LastUpdateTimestamp>
  <Topic>demo</Topic>
  <MaxOffset>0</MaxOffset>
  <MinOffset>0</MinOffset>
```

```

        </OffsetTable>
    </OffsetTable>
</TopicStatus>
<Success>true</Success>
<Code>200</Code>
</GetTopicStatusResponse>

```

JSON 格式

```

{
  "RequestId": "E475C7E2-8C35-46EF-BE7D-5D2A9F5D****",
  "Message": "operation success.",
  "TopicStatus": {
    "LastTimeStamp": "1566470063575",
    "TotalCount": "423",
    "OffsetTable": {
      "OffsetTable": [
        {
          "Partition": "0",
          "MaxOffset": "76",
          "Topic": "testkafka",
          "LastUpdateTimestamp": "1566470063547",
          "MinOffset": "0"
        }
      ]
    }
  },
  "Code": "200",
  "Success": "true"
}

```

### 错误码

访问[错误中心](#)查看更多错误码。

## 11.5. ModifyPartitionNum

调用ModifyPartitionNum修改Topic的分区数。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ModifyPartitionNum	要执行的操作。取值： <b>ModifyPartitionNum</b> 。

名称	类型	是否必选	示例值	描述
AddPartitionNum	Integer	是	6	增加分区数。 <ul style="list-style-type: none"> <li>必须大于0。</li> <li>建议分区数是6的倍数，减少数据倾斜风险。</li> <li>分区总数限制1~360。</li> <li>特殊需求请提交<a href="#">工单</a>。</li> </ul>
InstanceId	String	是	alikafka_post-cn-0pp1l9z8***	实例ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。
Topic	String	是	TopicPartitionNum	Topic名称。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success	返回信息。
RequestId	String	B7A39AE5-0B36-4442-A304-E0885265***	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=ModifyPartitionNum
&AddPartitionNum=6
&InstanceId=alikafka_post-cn-0pp1l9z8***
&RegionId=cn-hangzhou
&Topic=TopicPartitionNum
&<公共请求参数>
```

### 正常返回示例

XML 格式



```
<ModifyPartitionNumResponse>
  <RequestId>B7A39AE5-0B36-4442-A304-E0885265***</RequestId>
  <Message>operation success</Message>
  <Code>200</Code>
  <Success>>true</Success>
</ModifyPartitionNumResponse>
```

## JSON 格式

```
{
  "RequestId": "B7A39AE5-0B36-4442-A304-E0885265***",
  "Message": "operation success",
  "Code": 200,
  "Success": true
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 11.6. ModifyTopicRemark

调用ModifyTopicRemark修改Topic的备注。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ModifyTopicRemark	要执行的操作。取值： <b>ModifyTopicRemark</b> 。
InstanceId	String	是	alikafka_post-cn-0pp1l9z8***	实例ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。
Topic	String	是	alikafka_post-cn-0pp1l9z8***	Topic的名称。
Remark	String	否	testremark	备注。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success	返回信息。
RequestId	String	DB6F1BEA-903B-4FD8-8809-46E7E9CE***	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=ModifyTopicRemark
&InstanceId=alikafka_post-cn-0pp119z8***
&RegionId=cn-hangzhou
&Topic=alikafka_post-cn-0pp119z8z***
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<ModifyTopicRemarkResponse>
  <RequestId>DB6F1BEA-903B-4FD8-8809-46E7E9CE***</RequestId>
  <Message>operation success</Message>
  <Code>200</Code>
  <Success>>true</Success>
</ModifyTopicRemarkResponse>
```

#### JSON 格式

```
{
  "RequestId": "DB6F1BEA-903B-4FD8-8809-46E7E9CE***",
  "Message": "operation success",
  "Code": "200",
  "Success": "true"
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 12.Group

## 12.1. DeleteConsumerGroup

调用DeleteConsumerGroup删除Group。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteConsumerGroup	要执行的操作。取值： <b>DeleteConsumerGroup</b> 。
ConsumerId	String	是	CID-test	Group名称。
InstanceId	String	是	alikafka_post-cn-v0h1fgs2****	实例ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。

### 返回数据

名称	类型	示例值	描述
Code	Integer	200	状态码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	06084011-E093-46F3-A51F-4B19A8AD****	请求的ID。
Success	Boolean	true	调用是否成功。

### 示例

请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=DeleteConsumerGroup
&ConsumerId=CID-test
&InstanceId=alikafka_post-cn-v0h1fgs2****
&RegionId=cn-hangzhou
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DeleteConsumerGroupResponse>
  <Message>operation success.</Message>
  <RequestId>06084011-E093-46F3-A51F-4B19A8AD****</RequestId>
  <Success>true</Success>
  <Code>200</Code>
</DeleteConsumerGroupResponse>
```

#### JSON 格式

```
{
  "RequestId": "06084011-E093-46F3-A51F-4B19A8AD****",
  "Message": "operation success.",
  "Code": "200",
  "Success": "true"
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

## 12.2. GetConsumerList

调用GetConsumerList获取Group信息。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetConsumerList	系统规定参数。取值： <b>GetConsumerList</b> 。
InstanceId	String	是	alikafka_post-cn-v0h18sav****	Group所属实例的ID。
RegionId	String	是	cn-hangzhou	Group所属实例的地域ID。

名称	类型	是否必选	示例值	描述
ConsumerId	String	否	kafka-test	指定Group的ID。如果不指定，则查询所有Group。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	808F042B-CB9A-4FBC-9009-00E7DDB6****	请求ID。
Success	Boolean	true	调用是否成功。
ConsumerList	Array of ConsumerVO		Group列表。
ConsumerVO			
ConsumerId	String	kafka-test	Group名称。
Instanceld	String	alikaafka_post-cn-v0h18sav****	实例ID。
Remark	String	test	备注。
RegionId	String	cn-hangzhou	地域ID。
Tags	Array of TagVO		标签。
TagVO			
Key	String	test	标签键。
Value	String	test	标签值。

## 示例

请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=GetConsumerList
&InstanceId=alikafka_post-cn-v0h18sav****
&RegionId=cn-hangzhou
&ConsumerId=kafka-test
&公共请求参数
```

## 正常返回示例

### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<GetConsumerListResponse>
  <Code>200</Code>
  <Message>operation success.</Message>
  <RequestId>808F042B-CB9A-4FBC-9009-00E7DDB6****</RequestId>
  <Success>>true</Success>
  <ConsumerList>
    <ConsumerId>kafka-test</ConsumerId>
    <InstanceId>alikafka_post-cn-v0h18sav****</InstanceId>
    <Remark>test</Remark>
    <RegionId>cn-hangzhou</RegionId>
    <Tags>
      <Key>test</Key>
      <Value>test</Value>
    </Tags>
  </ConsumerList>
</GetConsumerListResponse>
```

### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : 200,
  "Message" : "operation success.",
  "RequestId" : "808F042B-CB9A-4FBC-9009-00E7DDB6****",
  "Success" : true,
  "ConsumerList" : [ {
    "ConsumerId" : "kafka-test",
    "InstanceId" : "alikafka_post-cn-v0h18sav****",
    "Remark" : "test",
    "RegionId" : "cn-hangzhou",
    "Tags" : [ {
      "Key" : "test",
      "Value" : "test"
    } ]
  } ]
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

## 12.3. GetConsumerProgress

调用GetConsumerProgress查询Group的消费状态。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetConsumerProgress	要执行的操作。取值： <b>GetConsumerProgress</b> 。
ConsumerId	String	是	kafka-test	Group的名称。
InstanceId	String	是	alikafka_pre-cn-mp919o4v****	实例ID。
RegionId	String	是	cn-hangzhou	实例的地域ID。

### 返回数据

名称	类型	示例值	描述
Code	Integer	200	状态码。返回200代表成功。
ConsumerProgress	Struct		消费状态。
LastTimestamp	Long	1566874931671	此Group最后被消费的消息的产生时间。
TopicList	Array of TopicList		此Group对应的每个Topic的消费进度列表。
TopicList			
LastTimestamp	Long	1566874931649	该Topic最后被消费的消息的产生时间。
OffsetList	Array of OffsetList		偏移列表。
OffsetList			

名称	类型	示例值	描述
BrokerOffset	Long	9	最大位点。
ConsumerOffset	Long	9	消费位点。
LastTimestamp	Long	1566874931649	该分区最后被消费的消息的产生时间。
Partition	Integer	0	分区ID。
Topic	String	kafka-test	Topic名称。
TotalDiff	Long	0	该Topic的未消费消息总量，即堆积量。
TotalDiff	Long	0	所有Topic的未消费消息总量，即堆积量。
Message	String	operation success.	返回信息。
RequestId	String	252820E1-A2E6-45F2-B4C9-1056B8CE****	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=GetConsumerProgress
&RegionId=cn-hangzhou
&ConsumerId=kafka-test
&InstanceId=alikafka_pre-cn-mp919o4v****
&<公共请求参数>
```

### 正常返回示例

XML 格式



```
<GetConsumerProgressResponse>
  <RequestId>252820E1-A2E6-45F2-B4C9-1056B8CE****</RequestId>
  <Message>operation success.</Message>
  <ConsumerProgress>
    <LastTimestamp>1566874931671</LastTimestamp>
    <TopicList>
      <TopicList>
        <OffsetList>
          <OffsetList>
            <Partition>0</Partition>
            <ConsumerOffset>9</ConsumerOffset>
            <LastTimestamp>1566874931671</LastTimestamp>
            <BrokerOffset>9</BrokerOffset>
          </OffsetList>
        </OffsetList>
      </TopicList>
    </TopicList>
    <TotalDiff>0</TotalDiff>
  </ConsumerProgress>
  <Code>200</Code>
  <Success>true</Success>
</GetConsumerProgressResponse>
```

JSON 格式

```

{
  "RequestId": "252820E1-A2E6-45F2-B4C9-1056B8CE****",
  "Message": "operation success.",
  "ConsumerProgress": {
    "LastTimestamp": 1566874931671,
    "TopicList": {
      "TopicList": [
        {
          "OffsetList": {
            "OffsetList": [
              {
                "Partition": 0,
                "ConsumerOffset": 9,
                "LastTimestamp": 1566874931671,
                "BrokerOffset": 9
              }
            ]
          },
          "LastTimestamp": 1566874931671,
          "TotalDiff": 0,
          "Topic": "kafka_test"
        }
      ]
    },
    "TotalDiff": 0
  },
  "Code": 200,
  "Success": true
}

```

## 错误码

访问[错误中心](#)查看更多错误码。

# 12.4. CreateConsumerGroup

调用CreateConsumerGroup创建Group。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateConsumerGroup	系统规定参数。取值：CreateConsumerGroup。
InstanceId	String	是	alikafka_post-cn-0pp1l9z8****	实例ID。

名称	类型	是否必选	示例值	描述
ConsumerId	String	是	test	Group名称。取值： <ul style="list-style-type: none"> <li>只能包含字母、数字、短划线 (-)、下划线 (_)，且至少包含一个英文或数字。</li> <li>长度限制在3~128字符，多于128字符将被自动截取。</li> <li>Group名称一旦创建后，将不能修改。</li> </ul>
RegionId	String	是	cn-hangzhou	实例的地域ID。
Remark	String	否	test	备注。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	状态码。返回200代表成功。
Message	String	operation success	返回信息。
RequestId	String	E57A8862-DF68-4055-8E55-B80CB4****	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=CreateConsumerGroup
&InstanceId=alikafka_post-cn-0pp119z8****
&ConsumerId=test
&RegionId=cn-hangzhou
&Remark=test
&公共请求参数
```

### 正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<CreateConsumerGroupResponse>
  <Code>200</Code>
  <Message>operation success</Message>
  <RequestId>E57A8862-DF68-4055-8E55-B80CB4****</RequestId>
  <Success>true</Success>
</CreateConsumerGroupResponse>
```

#### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "Code" : 200,
  "Message" : "operation success",
  "RequestId" : "E57A8862-DF68-4055-8E55-B80CB4****",
  "Success" : true
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 13. 标签

## 13.1. UntagResources

调用UntagResources为资源解绑标签。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UntagResources	系统规定参数。取值： <b>UntagResources</b> 。
RegionId	String	是	cn-hangzhou	资源的地域ID。
ResourceType	String	是	INSTANCE	资源类型。枚举类型，目前支持的资源类型： <ul style="list-style-type: none"> <li>• INSTANCE</li> <li>• TOPIC</li> <li>• CONSUMERGROUP</li> </ul> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <span style="font-size: 1em;">?</span> 说明 不区分大小写。         </div>
All	Boolean	否	false	是否解除资源绑定的全部标签。TagKey.N为空时，该参数有效。默认值为 <b>false</b> 。

名称	类型	是否必选	示例值	描述
ResourceId.N	String	是	alikafka_post-cn-v0h1fgs2****	<p>需要解绑的资源ID。资源ID规则：</p> <ul style="list-style-type: none"> <li>实例：instanceId</li> <li>Topic：Kafka_instanceId_topic</li> <li>Group：Kafka_instanceId_consumerGroup</li> </ul> <p>例如：需要解绑定的资源包括实例（alikafka_post-cn-v0h1fgs2xxxx）、Topic（test-topic）、Group（test-consumer-group），则各的资源ID分别为alikafka_post-cn-v0h1fgs2xxxx、Kafka_alikafka_post-cn-v0h1fgs2xxxx_test-topic、Kafka_alikafka_post-cn-v0h1fgs2xxxx_test-consumer-group。</p>
TagKey.N	String	否	FinanceDept	<p>资源的标签键。</p> <ul style="list-style-type: none"> <li>N为1~20。</li> <li>如果为空，且All为true时，匹配所有标签键。</li> <li>最多支持128个字符，不能以aliyun和acs:开头，不能包含http://或者https://。</li> </ul>

## 返回数据

名称	类型	示例值	描述
RequestId	String	C46FF5A8-C5F0-4024-8262-B16B639225A0	请求ID。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=UntagResources
&RegionId=cn-hangzhou
&ResourceId.1=alikafka_post-cn-v0h1fgs2****
&ResourceType=INSTANCE
&<公共请求参数>
```

### 正常返回示例

XML 格式

```

HTTP/1.1 200 OK
Content-Type:application/xml
<UntagResourcesResponse>
  <RequestId>C46FF5A8-C5F0-4024-8262-B16B639225A0</RequestId>
</UntagResourcesResponse>

```

JSON 格式

```

HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "C46FF5A8-C5F0-4024-8262-B16B639225A0"
}

```

### 错误码

访问[错误中心](#)查看更多错误码。

## 13.2. TagResources

调用TagResources为资源绑定标签。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	TagResources	系统规定参数。取值： <b>TagResources</b> 。
RegionId	String	是	cn-hangzhou	资源的地域ID。
ResourceType	String	是	instance	资源类型。枚举类型。取值： <ul style="list-style-type: none"> <li>INSTANCE</li> <li>TOPIC</li> <li>CONSUMERGROUP</li> </ul> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <span style="color: #00aaff;">?</span> 说明 不区分大小写。         </div>

名称	类型	是否必选	示例值	描述
ResourceId.N	String	是	alikafka_post-cn-v0h1fgs2****	<p>需要打标的资源ID。资源ID规则：</p> <ul style="list-style-type: none"> <li>实例：instanceld</li> <li>Topic：Kafka_alikafka_instanceld_topic</li> <li>Group：Kafka_alikafka_instanceld_consumer Group</li> </ul> <p>例如：实例ID为alikafka_post-cn-v0h1fgs2xxxx、Topic名称为test-topic、Group名称为test-consumer-group，则各资源ID分别为alikafka_post-cn-v0h1fgs2xxxx、Kafka_alikafka_post-cn-v0h1fgs2xxxx_test-topic、Kafka_alikafka_post-cn-v0h1fgs2xxxx_test-consumer-group。</p>
Tag.N.Key	String	是	FinanceDept	<p>资源的标签键。</p> <ul style="list-style-type: none"> <li>N为1~20。</li> <li>不允许为空。</li> <li>最多支持128个字符，不能以aliyun和acs:开头，不能包含http://或者https://。</li> </ul>
Tag.N.Value	String	否	FinanceJoshua	<p>资源的标签值。</p> <ul style="list-style-type: none"> <li>N为1~20。</li> <li>可以为空。</li> <li>最多支持128个字符，不能以aliyun和acs:开头，不能包含http://或者https://。</li> </ul>

## 返回数据

名称	类型	示例值	描述
RequestId	String	C46FF5A8-C5F0-4024-8262-B16B639225A0	请求ID。

## 示例

请求示例



```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=TagResources
&RegionId=cn-hangzhou
&ResourceId.1=alikafka_post-cn-v0h1fgs2****
&ResourceType=INSTANCE
&Tag.1.Key=FinanceDept
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<TagResourcesResponse>
  <RequestId>C46FF5A8-C5F0-4024-8262-B16B639225A0</RequestId>
</TagResourcesResponse>
```

#### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "C46FF5A8-C5F0-4024-8262-B16B639225A0"
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 13.3. ListTagResources


调用ListTagResources查询资源绑定的标签列表。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListTagResources	要执行的操作。取值： <b>ListTagResources</b> 。
RegionId	String	是	cn-hangzhou	资源的地域ID。

名称	类型	是否必选	示例值	描述
ResourceType	String	是	instance	资源类型。枚举类型。取值： <ul style="list-style-type: none"> <li>INSTANCE</li> <li>TOPIC</li> <li>CONSUMERGROUP</li> </ul>
ResourceId.N	RepeatList	否	alikafka_post-cn-v0h1fgs2****	<p>打标的资源ID。资源ID规则：</p> <ul style="list-style-type: none"> <li>实例：instanceld</li> <li>Topic：Kafka_alikafka_instanceld_topic</li> <li>Group：Kafka_alikafka_instanceld_consumer Group</li> </ul> <p>例如：实例ID为alikafka_post-cn-v0h1fgs2xxxx、Topic名称为test-topic、Group名称为test-consumer-group，则各资源ID分别为alikafka_post-cn-v0h1fgs2xxxx、Kafka_alikafka_post-cn-v0h1fgs2xxxx_test-topic、Kafka_alikafka_post-cn-v0h1fgs2xxxx_test-consumer-group。</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> <b>说明</b> ResourceId与Tag参数必须配置其中任意一个，才能查询资源绑定的标签列表。否则，API调用失败。</p> </div>
Tag.N.Key	String	否	FinanceDept	<p>资源的标签键。</p> <ul style="list-style-type: none"> <li>N为1~20。</li> <li>如果为空，则匹配所有标签键。</li> <li>最多支持128个字符，不能以aliyun和acs:开头，不能包含http://或者https://。</li> </ul>
Tag.N.Value	String	否	FinanceJoshua	<p>资源的标签值。</p> <ul style="list-style-type: none"> <li>N为1~20。</li> <li>如果标签键为空，则必须为空。为空时，匹配所有标签值。</li> <li>最多支持128个字符，不能以aliyun和acs:开头，不能包含http://或者https://。</li> </ul>

名称	类型	是否必选	示例值	描述
NextToken	String	否	caeba0bbb2be03f84eb48b699f0a4883	下一个查询开始的Token。

## 返回数据

名称	类型	示例值	描述
NextToken	String	caeba0bbb2be03f84eb48b699f0a4883	下一个查询开始Token。
RequestId	String	DE65F6B7-7566-4802-9007-96F2494AC5XX	请求ID。
TagResources	Array of TagResource		由资源及其标签组成的集合，包含了资源ID、资源类型和标签键值等信息。
TagResource			
ResourceId	String	alikafka_post-cn-v0h1fgs2****	<p>打标的资源ID。资源ID规则：</p> <ul style="list-style-type: none"> <li>实例：instanceId</li> <li>Topic：Kafka_alikafka_instanceId_topic</li> <li>Consumer Group：Kafka_alikafka_instanceId_consumerGroup</li> </ul> <p>例如：实例ID为alikafka_post-cn-v0h1fgs2xxxx、Topic名称为test-topic、Consumer Group名称为test-consumer-group，则各资源ID分别为alikafka_post-cn-v0h1fgs2xxxx、Kafka_alikafka_post-cn-v0h1fgs2xxxx_test-topic、Kafka_alikafka_post-cn-v0h1fgs2xxxx_test-consumer-group。</p>
ResourceType	String	instance	<p>资源类型。枚举类型。取值：</p> <ul style="list-style-type: none"> <li><b>Instance</b></li> <li><b>Topic</b></li> <li><b>Consumergroup</b></li> </ul>
TagKey	String	FinanceDept	标签键。
TagValue	String	FinanceJoshua	标签值。

## 示例

## 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=ListTagResources
&RegionId=cn-hangzhou
&ResourceType=INSTANCE
&ResourceId=alikafka_post-cn-v0h1fgs2****
&<公共请求参数>
```

## 正常返回示例

### XML 格式

```
<ListTagResourcesResponse>
  <RequestId>DE65F6B7-7566-4802-9007-96F2494AC5XX</RequestId>
  <NextToken>caeba0bbb2be03f84eb48b699f0a4883</NextToken>
  <TagResources>
    <TagResource>
      <ResourceId>alikafka_post-cn-v0h1fgs2****</ResourceId>
      <TagKey>FinanceDept</TagKey>
      <ResourceType>ALIYUN::ALIKAFKA::INSTANCE</ResourceType>
      <TagValue>FinanceJoshua</TagValue>
    </TagResource>
  </TagResources>
</ListTagResourcesResponse>
```

### JSON 格式

```
{
  "RequestId": "DE65F6B7-7566-4802-9007-96F2494AC5XX",
  "NextToken": "caeba0bbb2be03f84eb48b699f0a4883",
  "TagResources": {
    "TagResource": [
      {
        "ResourceId": "alikafka_post-cn-v0h1fgs2****",
        "TagKey": "FinanceDept",
        "ResourceType": "ALIYUN::ALIKAFKA::INSTANCE",
        "TagValue": "FinanceJoshua"
      }
    ]
  }
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 14.SASL用户

## 14.1. CreateSaslUser

调用CreateSaslUser创建SASL用户。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateSaslUser	要执行的操作。取值： <b>CreateSaslUser</b> 。
InstanceId	String	是	alikafka_pre-cn-v0h1cng0****	实例ID。
Password	String	是	12***	密码。
RegionId	String	是	cn-hangzhou	地域ID。
Username	String	是	test***	用户名。
Type	String	否	plain	类型。取值： <ul style="list-style-type: none"><li>• <b>plain</b>：一种简单的用户名密码校验机制。消息队列Kafka版优化了PLAIN机制，支持不重启实例的情况下动态增加SASL用户。</li><li>• <b>scram</b>：一种用户名密码校验机制，安全性比PLAIN更高。消息队列Kafka版使用SCRAM-SHA-256。</li></ul> 默认值为 <b>plain</b> 。

### 返回数据

名称	类型	示例值	描述
Code	Integer	200	状态码。返回200代表成功。

名称	类型	示例值	描述
Message	String	operation success.	返回消息。
RequestId	String	C5CA600C-7D5A-45B5-B6DB-44FAC2C****	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=CreateSaslUser
&InstanceId=alikafka_pre-cn-v0h1cng0****
&Password=12***
&RegionId=cn-hangzhou
&Username=test***
&<公共请求参数>
```

### 正常返回示例

XML 格式

```
<CreateSaslUserResponse>
  <RequestId>C5CA600C-7D5A-45B5-B6DB-44FAC2C****</RequestId>
  <Message>operation success</Message>
  <Code>200</Code>
  <Success>>true</Success>
</CreateSaslUserResponse>
```

JSON 格式

```
{
  "RequestId": "C5CA600C-7D5A-45B5-B6DB-44FAC2C****",
  "Message": "operation success.",
  "Code": "200",
  "Success": "true"
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 14.2. DeleteSaslUser

调用DeleteSaslUser删除SASL用户。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteSaslUser	要执行的操作。取值： <b>DeleteSaslUser</b> 。
InstanceId	String	是	alikaafka_pre-cn-v0h1cng0****	实例ID。
RegionId	String	是	cn-hangzhou	地域ID。
Username	String	是	test***	用户名。
Type	String	否	scram	类型。取值： <ul style="list-style-type: none"> <li><b>plain</b>：一种简单的用户名密码校验机制。消息队列Kafka版优化了PLAIN机制，支持不重启实例的情况下动态增加SASL用户。</li> <li><b>scram</b>：一种用户名密码校验机制，安全性比PLAIN更高。消息队列Kafka版使用SCRAM-SHA-256。</li> </ul> 默认值为 <b>plain</b> 。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	状态码。返回200代表成功。
Message	String	operation success	返回信息。
RequestId	String	3CB89F5C-CD97-4C1D-BC7C-FEDEC2F4****	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

请求示例

```
http(s)://[Endpoint]/?Action=DeleteSaslUser
&InstanceId=alikafka_pre-cn-v0h1cng0****
&RegionId=cn-hangzhou
&Username=test***
&Type=scram
&<公共请求参数>
```

### 正常返回示例

XML 格式

```
<DeleteSaslUserResponse>
  <RequestId>3CB89F5C-CD97-4C1D-BC7C-FEDEC2F4****</RequestId>
  <Message>operation success</Message>
  <Code>200</Code>
  <Success>>true</Success>
</DeleteSaslUserResponse>
```

JSON 格式

```
{
  "RequestId": "3CB89F5C-CD97-4C1D-BC7C-FEDEC2F4****",
  "Message": "operation success",
  "Code": 200,
  "Success": true
}
```

### 错误码

访问[错误中心](#)查看更多错误码。

## 14.3. DescribeSaslUsers

调用DescribeSaslUsers查询SASL用户。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeSaslUsers	要执行的操作。取值： <b>DescribeSaslUsers</b> 。
InstanceId	String	是	alikafka_pre-cn-v0h1cng0****	实例ID。



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

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	状态码。返回200代表成功。
Message	String	operation success.	返回信息。
RequestId	String	9E3B3592-5994-4F65-A61E-E62A77A7***	请求ID。
SaslUserList	Array of SaslUserVO		SASL用户列表。
SaslUserVO			
Password	String	123***	密码。
Type	String	scram	类型。取值： <ul style="list-style-type: none"> <li>• <b>PLAIN</b>：一种简单的用户名密码校验机制。消息队列Kafka版优化了PLAIN机制，支持不重启实例的情况下动态增加SASL用户。</li> <li>• <b>SCRAM</b>：一种用户名密码校验机制，安全性比PLAIN更高。消息队列Kafka版使用SCRAM-SHA-256。</li> </ul> 默认值为 <b>plain</b> 。
Username	String	test12***	用户名。
Success	Boolean	true	是否成功。

## 示例

### 请求示例

```

http(s)://[Endpoint]/?Action=DescribeSaslUsers
&InstanceId=alikafka_pre-cn-v0h1cng0****
&RegionId=cn-hangzhou
&<公共请求参数>

```

## 正常返回示例

### XML 格式

```
<DescribeSaslUsersResponse>
  <SaslUserList>
    <SaslUserVO>
      <Type>scram</Type>
      <Username>test12***</Username>
      <Password>123***</Password>
    </SaslUserVO>
  </SaslUserList>
  <RequestId>9E3B3592-5994-4F65-A61E-E62A77A7***</RequestId>
  <Message>operation success.</Message>
  <Code>200</Code>
  <Success>>true</Success>
</DescribeSaslUsersResponse>
```

### JSON 格式

```
{
  "SaslUserList": {
    "SaslUserVO": [
      {
        "Type": "scram",
        "Username": "test12***",
        "Password": "123***"
      }
    ]
  },
  "RequestId": "9E3B3592-5994-4F65-A61E-E62A77A7***",
  "Message": "operation success.",
  "Code": 200,
  "Success": true
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 15.SASL ACL

## 15.1. CreateAcl

调用CreateAcl创建ACL。

### 调试

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

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateAcl	要执行的操作。取值： <b>CreateAcl</b> 。
AclOperationType	String	是	Read	操作类型。取值： <ul style="list-style-type: none"> <li><b>Write</b>：写入。</li> <li><b>Read</b>：读取。</li> <li><b>Describe</b>：读取TransactionalId。</li> <li><b>IdempotentWrite</b>：幂等写Cluster。</li> </ul>
AclResourceName	String	是	x***	资源名称。 <ul style="list-style-type: none"> <li>资源的名称，可以是Topic名称、Group ID、集群名称或事务ID。</li> <li>支持使用星号（*）表示该类型的资源。</li> </ul>
AclResourcePatternType	String	是	LITERAL	匹配模式。取值： <ul style="list-style-type: none"> <li><b>LITERAL</b>：全匹配</li> <li><b>PREFIXED</b>：前缀匹配</li> </ul>
AclResourceType	String	是	Group	资源类型。取值： <ul style="list-style-type: none"> <li><b>Topic</b></li> <li><b>Group</b></li> <li><b>Cluster</b></li> <li><b>TransactionalId</b></li> </ul>
InstanceId	String	是	alikafka_pre-cn-v0h1cng00***	实例ID。

名称	类型	是否必选	示例值	描述
RegionId	String	是	cn-hangzhou	地域ID。
Username	String	是	test***	用户名。 支持使用星号（*）表示所有用户名。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	状态码。返回200代表成功。
Message	String	operation success	返回信息。
RequestId	String	56729737-C428-4E1B-AC68-7A8C2D5****	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=CreateAcl
&AclOperationType=Read
&AclResourceName=X***
&AclResourcePatternType=LITERAL
&AclResourceType=Group
&InstanceId=alikafka_pre-cn-v0h1cng00***
&RegionId=cn-hangzhou
&Username=test***
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<CreateAclResponse>
  <RequestId>56729737-C428-4E1B-AC68-7A8C2D5****</RequestId>
  <Message>operation success</Message>
  <Code>200</Code>
  <Success>>true</Success>
</CreateAclResponse>
```

#### JSON 格式

```
{
  "RequestId": "56729737-C428-4E1B-AC68-7A8C2D5****",
  "Message": "operation success",
  "Code": 200,
  "Success": true
}
```

## 错误码

访问[错误中心](#)查看更多错误码。

# 15.2. DescribeAcls

调用DescribeAcls查询ACL。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeAcls	要执行的操作。取值： <b>DescribeAcls</b> 。
AclResourceName	String	是	demo	资源名称。 <ul style="list-style-type: none"> <li>Topic或Group的名称。</li> <li>支持使用星号（*）表示所有Topic或Group的名称。</li> </ul>
AclResourceType	String	是	Topic	资源类型。取值： <ul style="list-style-type: none"> <li><b>Topic</b></li> <li><b>Group</b></li> </ul>
InstanceId	String	是	alibaba_pre-cn-v0h1cng0***	实例ID。
RegionId	String	是	cn-hangzhou	地域ID。
Username	String	是	test12****	用户名。

名称	类型	是否必选	示例值	描述
AclResourcePatternType	String	否	LITERAL	匹配模式。取值： <ul style="list-style-type: none"> <li>LITERAL：全匹配</li> <li>PREFIXED：前缀匹配</li> </ul>

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	状态码。返回200代表成功。
KafkaAclList	Array of KafkaAclVO		ACL列表。
KafkaAclVO			
AclOperationType	String	Write	操作类型。取值： <ul style="list-style-type: none"> <li><b>Write</b>：写入</li> <li><b>Read</b>：读取</li> </ul>
AclResourceName	String	demo	资源名称。 <ul style="list-style-type: none"> <li>Topic或Consumer Group的名称。</li> <li>支持使用星号（*）表示所有Topic或Consumer Group的名称。</li> </ul>
AclResourcePatternType	String	LITERAL	匹配模式。取值： <ul style="list-style-type: none"> <li>LITERAL：全匹配</li> <li>PREFIXED：前缀匹配</li> </ul>
AclResourceType	String	Topic	资源类型。取值： <ul style="list-style-type: none"> <li><b>Topic</b></li> <li><b>Group</b></li> </ul>
Host	String	*	主机。
Username	String	test12***	用户名。
Message	String	operation success.	返回信息。

名称	类型	示例值	描述
RequestId	String	46496E38-881E-4719-A2F3-F3DA6AEA***	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://alikafka.cn-hangzhou.aliyuncs.com/?Action=DescribeAcls
&AclResourceName=demo
&AclResourceType=Topic
&InstanceId=alikafka_pre-cn-v0h1cng0***
&RegionId=cn-hangzhou
&Username=test12****
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DescribeAclsResponse>
  <RequestId>46496E38-881E-4719-A2F3-F3DA6AEA***</RequestId>
  <Message>operation success.</Message>
  <Code>200</Code>
  <Success>>true</Success>
  <KafkaAclList>
    <KafkaAclVO>
      <AclResourceName>demo</AclResourceName>
      <Username>test12***</Username>
      <AclResourceType>Topic</AclResourceType>
      <AclOperationType>Write</AclOperationType>
      <AclResourcePatternType>LITERAL</AclResourcePatternType>
      <Host>*</Host>
    </KafkaAclVO>
  </KafkaAclList>
</DescribeAclsResponse>
```

#### JSON 格式

```

{
  "RequestId": "46496E38-881E-4719-A2F3-F3DA6AEA***",
  "Message": "operation success.",
  "Code": 200,
  "Success": true,
  "KafkaAclList": {
    "KafkaAclVO": [
      {
        "AclResourceName": "demo",
        "Username": "test12***",
        "AclResourceType": "Topic",
        "AclOperationType": "Write",
        "AclResourcePatternType": "LITERAL",
        "Host": "*"
      }
    ]
  }
}

```

## 错误码

访问[错误中心](#)查看更多错误码。

# 15.3. DeleteAcl

调用DeleteAcl删除ACL。

## 调试

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

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteAcl	要执行的操作。取值： <b>DeleteAcl</b> 。
AclOperationType	String	是	Write	操作类型。取值： <ul style="list-style-type: none"> <li><b>Write</b>：读取</li> <li><b>Read</b>：写入</li> </ul>
AclResourceName	String	是	demo	资源名称。 <ul style="list-style-type: none"> <li>Topic名称或Group的名称。</li> <li>星号（*）表示所有Topic或Group的名称。</li> </ul>



名称	类型	是否必选	示例值	描述
AclResourcePatternType	String	是	LITERAL	匹配模式。取值： <ul style="list-style-type: none"> <li>LITERAL：全匹配模式</li> <li>PREFIXED：前缀匹配模式</li> </ul>
AclResourceType	String	是	Topic	资源类型。 <ul style="list-style-type: none"> <li>Topic</li> <li>Group</li> </ul>
InstanceId	String	是	alibabacloud_pre-cn-v0h1cng0****	实例ID。
RegionId	String	是	cn-hangzhou	地域ID。
Username	String	是	test12****	用户名。

## 返回数据

名称	类型	示例值	描述
Code	Integer	200	状态码。返回200代表成功。
Message	String	operation success	返回信息。
RequestId	String	B0740227-AA9A-4E14-8E9F-36ED6652****	请求ID。
Success	Boolean	true	调用是否成功。

## 示例

### 请求示例

```
http(s)://alibabacloud_pre-cn-hangzhou.aliyuncs.com/?Action=DeleteAcl
&AclOperationType=Write
&AclResourceName=demo
&AclResourcePatternType=LITERAL
&AclResourceType=Topic
&InstanceId=alibabacloud_pre-cn-v0h1cng0****
&RegionId=cn-hangzhou
&Username=test12****
&<公共请求参数>
```

## 正常返回示例

### XML 格式

```
<DeleteAclResponse>
  <RequestId>B0740227-AA9A-4E14-8E9F-36ED6652***</RequestId>
  <Message>operation success</Message>
  <Code>200</Code>
  <Success>>true</Success>
</DeleteAclResponse>
```

### JSON 格式

```
{
  "RequestId": "B0740227-AA9A-4E14-8E9F-36ED6652***",
  "Message": "operation success",
  "Code": 200,
  "Success": true
}
```

## 错误码

访问[错误中心](#)查看更多错误码。