

# 阿里云 应用配置管理 ACM

API 参考

文档版本：20191216

# 法律声明

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

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或惩罚性的损害，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. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，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。未经阿里云和/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
	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。	禁止： 重置操作将丢失用户配置数据。
	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。
	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。
	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，单击确定。
<b>Courier字体</b>	命令。	执行cd /d C:/window命令，进入Windows系统文件夹。
<code>##</code>	表示参数、变量。	<code>bae log list --instanceid Instance_ID</code>
<code>[]或者[a b]</code>	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
<code>{}</code> 或者 <code>{a b}</code>	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

---

法律声明.....	I
通用约定.....	I
1 API 简介.....	1
2 API 概览.....	2
3 getConfig.....	6
4 getAllConfigByTenant.....	8
5 addListener.....	11
6 syncUpdateAll.....	14
7 deleteAllDatums.....	17

# 1 API 简介

---

您可以使用本文档介绍的 API 对应用配置服务进行相关操作。

目前提供配置查询接口和监听相关接口。请确保在使用这些接口前，您已充分了解 ACM 产品说明和使用协议。

## 2 API 概览

本文介绍 ACM API 的概要信息，包括 API 列表、获取服务器 IP 的方法、通信协议、请求方法、公共参数、签名算法等。

### 配置管理 API

API	描述
<code>getConfig</code>	获取 ACM 配置内容。
<code>getAllConfigByTenant</code>	获取指定命名空间内的 ACM 配置信息。
<code>addListener</code>	监听 ACM 配置的变更。
<code>syncUpdateAll</code>	发布 ACM 配置。
<code>deleteAllDatums</code>	删除 ACM 配置。

### 获取服务器 IP 列表

通过地址服务器获取服务 IP 列表，以便通过服务 IP 发起请求。

```
http://${地址服务器域名}:8080/diamond-server/diamond
```

表 2-1: 地址服务器域名

地域	地址服务器域名
公网	acm.aliyun.com
华东 1 (杭州)	addr-hz-internal.edas.aliyun.com
华北 1 (青岛)	addr-qd-internal.edas.aliyun.com
华东 2 (上海)	addr-sh-internal.edas.aliyun.com
华北 2 (北京)	addr-bj-internal.edas.aliyun.com
华南 1 (深圳)	addr-sz-internal.edas.aliyun.com
中国 (香港)	addr-hk-internal.edas.aliyuncs.com
新加坡 (新加坡)	addr-singapore-internal.edas.aliyun.com
澳大利亚 (悉尼)	addr-ap-southeast-2-internal.edas.aliyun.com
美国 (硅谷)	addr-us-west-1-internal.acm.aliyun.com
美国 (弗吉尼亚)	addr-us-east-1-internal.acm.aliyun.com
华东 2 (上海) 金融云	addr-cn-shanghai-finance-1-internal.edas.aliyun.com

### 代码示例：

```
curl http://acm.aliyun.com:8080/diamond-server/diamond
# 返回
139.196.135.144
```

### 通信协议

**支持通过 HTTP 进行请求通信。**

### 请求方法

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

### 请求参数

**每个请求都需要包含公共的鉴权、签名相关请求参数和相关操作所特有的请求参数。**

### 公共参数

调用 ACM API 时都会用到的 Header 参数如下表所示。

参数	类型	是否必需	描述
Spas-AccessKey	String	是	在 ACM 控制台上的命名空间详情对话框内可获取 AccessKey。
timeStamp	String	是	以毫秒为单位的请求时间。
Spas-Signature	String	是	使用 SecretKey 对 “Tenant +TimeStamp” 签名 (SpasSigner .sign(tenant+ timeStamp , secretKey))，签名算法为 HmacSHA1。TimeStamp 签名的作用是防止重放攻击。该签名有效期为 60 秒。
Spas-SecurityToken	String	否	SecurityToken 需从 STS 临时凭证中获取。STS 临时凭证需从实例元数据 URL 中获取。详情请参考： <ul style="list-style-type: none"> <li>· 借助于实例 RAM 角色访问其他云产品</li> <li>· 通过 ECS 实例 RAM 角色访问 ACM</li> </ul>
longPollingTimeout	String	是	长轮询等待 30 秒，此处填写 30000。

### 字符编码

**请求及返回结果都使用 GBK 字符集进行编码。**

## 签名机制

ACM 服务会对每个访问的请求进行身份验证，使用 HTTP 需要在请求中包含签名（Signature）信息。ACM 通过使用 AccessKey 和 SecretKey 进行对称加密的方法来验证请求的发送者身份。

AccessKey 和 SecretKey 由 ACM 颁发给访问者。其中 AccessKey 用于标识访问者的身份，SecretKey 是用于加密签名字字符串和服务器端验证签名字字符串的密钥，出于安全考虑，请务必严格保密。

## 签名算法

签名采用 HmacSHA1 算法。

### · Java 签名算法参考

```
public static void main(String[] args) throws Exception {
    String tenant= "tenant";
    String group = "group";
    String timeStamp = String.valueOf(System.currentTimeMillis());
    String abc = HmacSHA1Encrypt(tenant+ "+" + group + "+" +
timeStamp , "1234");
    System.out.println(abc);
}
public static String HmacSHA1Encrypt(String encryptText, String
encryptKey) throws Exception {
    byte[] data = encryptKey.getBytes("UTF-8");
    // 根据给定的字节数组构造一个密钥，第二参数指定一个密钥算法的名称
    SecretKey secretKey = new SecretKeySpec(data, "HmacSHA1");
    // 生成一个指定 Mac 算法的 Mac 对象
    Mac mac = Mac.getInstance("HmacSHA1");
    // 用给定密钥初始化 Mac 对象
    mac.init(secretKey);
    byte[] text = encryptText.getBytes("UTF-8");
    byte[] textFinal = mac.doFinal(text);
    // 完成 Mac 操作，base64 编码，将 byte 数组转换为字符串
    return new String(Base64.encodeBase64(textFinal));
}
```

### · Shell 签名算法

```
## config sign
timestamp=`echo ${$(date +%s%N)/1000000}`
signStr=$namespace+$group+$timestamp
signContent=`echo -n $signStr | openssl dgst -hmac $sk -sha1 -binary
| base64`
echo $signContent
```

## 签名处理步骤

1. 使用请求参数构造规范的请求字符串（QueryParam）。

## 2. 使用上一步构造的规范字符串，按照以下规则构造用于计算签名的字符串。

Signature=HMAC-SHA1(QueryParam)



说明：

对于不同的请求，QueryParam 会不同。

## 3. 按照 RFC2104 的定义，使用上一步构造的用于签名的字符串来计算签名 HMAC 值。



说明：

计算签名时使用的 Key 就是您持有的 AccessKeySecret (ASCII:38)，使用的哈希算法是 SHA1。

## 4. 按照 Base64 编码规则将上一步算出的 HMAC 值编码成字符串，即得到签名值 (Signature)。

## 5. 将上一步得到的签名值作为 Signature 参数添加到请求参数中，即完成对请求的签名处理。

示例代码

以下示例代码展示了如何以 Shell 构造 ACM 请求。

```
#!/bin/bash
## config param
dataId="com.alibaba.nacos.example.properties"
group="DEFAULT_GROUP"
namespace="04754ad1-4f67-4d67-b2bf-1f73a04a****"
accessKey="8c5cbb849ae04682ad9f455a96aa****"
secretKey="lw05T7vfPJU27Fc1Pa+/CyIG****"
endpoint="acm.aliyun.com"
## config param end
## get serverIp from address server
serverIp=`curl $endpoint:8080/diamond-server/diamond -s | awk '{a[NR] = $0}END{ srand(); i=int(rand()*NR+1); print a[i]}'`
## config sign
timestamp=`echo ${$(date +%s%N)/1000000}`
signStr=$namespace+$group+$timestamp
signContent=`echo -n $signStr | openssl dgst -hmac $secretKey -sha1 -binary | base64`
## request to get a config
curl -H "Spas-AccessKey:$accessKey -H "timeStamp:$timestamp -H "
Spas-Signature:$signContent "http://"$serverIp":8080/diamond-server/
config.co?dataId="$dataId"&group="$group"&tenant="$namespace -v
```

限流机制

ACM 对访问频率采取限制，主要规则如下：

- 每个 IP 长连接数最多为 30 个。
- 每个 IP 每秒修改同一个配置不能超过 5 次。
- 每个 IP 每秒获取同一个配置不能超过 10 次。

## 3 getConfig

使用 getConfig 接口获取 ACM 配置。

请求类型

GET

请求 URL

*/diamond-server/config.co*

请求参数

参数	类型	是否必需	描述
tenant	String	是	租户信息，对应 ACM 的命名空间 ID。
dataId	String	是	配置的 ID
group	String	是	配置的分组

Header 参数

参数	类型	是否必需	描述
Spas-AccessKey	String	是	在 ACM 控制台上的命名空间详情对话框内可获取 AccessKey。
timeStamp	String	是	以毫秒为单位的请求时间。
Spas-Signature	String	是	使用 SecretKey 对 “Tenant +TimeStamp” 签名 (SpasSigner.sign(tenant+ timeStamp, secretKey))，签名算法为 HmacSHA1。TimeStamp 签名的作用是防止重放攻击。该签名有效期为 60 秒。
Spas-SecurityToken	String	否	SecurityToken 需从 STS 临时凭证中获取。STS 临时凭证需从实例元数据 URL 中获取。详情请参考： <ul style="list-style-type: none"> <li>· 借助于实例 RAM 角色访问其他云产品</li> <li>· 通过 ECS 实例 RAM 角色访问 ACM</li> </ul>

返回参数

参数	类型	描述
-	String	配置内容

## 错误码

错误码	错误信息	描述
400	Bad Request	客户端请求中的语法错误
403	Forbidden	没有权限
404	Not Found	客户端错误，未找到。
500	Internal Server Error	服务器内部错误

## 代码示例

### · 请求示例（Shell）

```
#!/bin/bash
## config param
dataId="com.alibaba.nacos.example.properties"
group="DEFAULT_GROUP"
namespace="04754ad1-4f67-4d67-b2bf-1f73a04a****"
accessKey="8c5cbb849ae04682ad9f455a96aa****"
secretKey="lw05T7vfPJU27FclPa+/CyIG****"
endpoint="acm.aliyun.com"
## config param end
## get serverIp from address server
serverIp=`curl $endpoint:8080/diamond-server/diamond -s | awk '{a[NR] = $0}END{ srand(); i=int(rand()*NR+1); print a[i]}'`
## config sign
timestamp=`echo ${$(date +%s%N)}/1000000` 
signStr=$namespace+$group+$timestamp
signContent=`echo -n $signStr | openssl dgst -hmac $secretKey -sha1
-binary | base64` 
## request to get a config
curl -H "Spas-AccessKey:$accessKey" -H "timeStamp:$timestamp" -H "
Spas-Signature:$signContent" "http://"$serverIp":8080/diamond-server
/config.co?dataId="$dataId"&group="$group"&tenant="$namespace" -v
```



#### 说明:

由于 shell 脚本在 Windows 系统下编辑容易编码错误，建议在 Linux 系统里新建 shell 脚本文件，再将此段代码复制到文件中。

### · 返回示例

```
connectTimeoutInMills=3000
```

## 更多信息

### [API 概览](#)

## 4 getAllConfigByTenant

使用 getAllConfigByTenant 接口获取指定命名空间内的 ACM 配置信息。

请求类型

GET

请求 URL

*/diamond-server/basestone.do?method=getAllConfigByTenant*

请求参数

参数	类型	是否必需	描述
tenant	String	是	租户信息，对应 ACM 的命名空间 ID。
pageNo	Integer	是	分页页号
pageSize	Integer	是	分页大小

Header 参数

参数	类型	是否必需	描述
Spas-AccessKey	String	是	在 ACM 控制台上的命名空间详情对话框内可获取 AccessKey。
timeStamp	String	是	以毫秒为单位的请求时间。
Spas-Signature	String	是	使用 SecretKey 对 “Tenant +TimeStamp” 签名 (SpasSigner.sign(tenant+ timeStamp, secretKey))，签名算法为 HmacSHA1。TimeStamp 签名的作用是防止重放攻击。该签名有效期为 60 秒。
Spas-SecurityToken	String	否	SecurityToken 需从 STS 临时凭证中获取。STS 临时凭证需从实例元数据 URL 中获取。详情请参考： <ul style="list-style-type: none"> <li>· 借助于实例 RAM 角色访问其他云产品</li> <li>· 通过 ECS 实例 RAM 角色访问 ACM</li> </ul>

返回参数

参数	类型	描述
totalCount	Integer	总配置数

参数	类型	描述
pageNumber	Integer	分页页号
pagesAvailable	Integer	可用分页数
pageItems	Array	配置信息
appName	String	归属应用的名称
dataId	String	配置的 ID
group	String	配置的分组

错误码

错误码	错误信息	描述
400	Bad Request	客户端请求中的语法错误
403	Forbidden	没有权限
404	Not Found	客户端错误，未找到。
500	Internal Server Error	服务器内部错误

代码示例

#### · 请求示例（Shell）

```
#!/bin/bash
## config param
namespace="04754ad1-4f67-4d67-b2bf-1f73a04a****"
accessKey="8c5cbb849ae04682ad9f455a96aa****"
secretKey="lw05T7vfPJU27FclPa+/CyIG****"
endpoint="acm.aliyun.com"
## config param end
## get serverIp from address server
serverIp=`curl $endpoint:8080/diamond-server/diamond -s | awk '{a[NR] = $0}END{rand();i=int(rand()*NR+1);print a[i]}'`
## config sign
timestamp=`echo ${$(date +%s%N)}/1000000`'
signStr=$namespace+$timestamp
signContent=`echo -n $signStr | openssl dgst -hmac $secretKey -sha1
-binary | base64`'
## request to get configs in a namespace
curl -H "Spas-AccessKey:$accessKey" -H "timeStamp:$timestamp" -H "
Spas-Signature:$signContent" "http://"$serverIp":8080/diamond-server
/basestone.do?method=getAllConfigByTenant&tenant="$namespace"&pageNo
="1"&pageSize="10" -v
```



说明：

由于 shell 脚本在 Windows 系统下编辑容易编码错误，建议在 Linux 系统里新建 shell 脚本文件，再将此段代码复制到文件中。

- 返回示例（JSON）

```
{  
    "totalCount":4,  
    "pageNumber":1,  
    "pagesAvailable":1,  
    "pageItems": [  
        {  
            "appName": "",  
            "dataId": "com.alibaba.nacos.example01.properties",  
            "group": "DEFAULT_GROUP"  
        },  
        {  
            "appName": "",  
            "dataId": "com.alibaba.nacos.example02.properties",  
            "group": "DEFAULT_GROUP"  
        },  
        {  
            "appName": "",  
            "dataId": "com.alibaba.nacos.example03.properties",  
            "group": "DEFAULT_GROUP"  
        },  
        {  
            "appName": "",  
            "dataId": "com.alibaba.nacos.example04.properties",  
            "group": "DEFAULT_GROUP"  
        }  
    ]  
}
```

更多信息

[API 概览](#)

# 5 addListener

---

使用 addListener 接口监听 ACM 配置的变更。

## API 描述

addListener 接口可监听 ACM 上的配置，并实时感知配置变更。如果配置变更，则您可以用 [getConfig](#) 接口获取配置的最新值，并动态刷新本地缓存。

注册监听采用的是异步 Servlet 技术。注册监听的本质是将配置和配置值的 MD5 值与后台对比，如果 MD5 值不一致，则立即返回不一致的配置。如果 MD5 值一致，则等待 30 秒，且返回值为空。

## 请求类型

**POST**

## 请求 URL

/diamond-server/config.co

## 请求参数

参数	类型	是否必需	描述
Probe-Modify-Request	String	是	<p>监听数据报文，格式为 <code>dataId^2group^2contentMD5^2tenant^1</code>。</p> <ul style="list-style-type: none"> <li>配置的多个字段间分隔符：(Java) <code>^2 = Character.toString((char) 2)</code> 或 (Shell) <code>^2=%02</code>。</li> <li>配置间分隔符：(Java) <code>^1 = Character.toString((char) 1)</code> 或 (Shell) <code>^1=%01</code></li> <li><b>contentMD5:</b> MD5(content)。第一次本地缓存为空，所以为空串。</li> </ul>

## Header 参数

参数	类型	是否必需	描述
Spas-AccessKey	String	是	在 ACM 控制台上的命名空间详情对话框内可获取 AccessKey。
timeStamp	String	是	以毫秒为单位的请求时间。

参数	类型	是否必需	描述
Spas-Signature	String	是	使用 SecretKey 对 “Tenant +TimeStamp” 签名 (SpasSigner.sign(tenant+ timeStamp, secretKey))，签名算法为 HmacSHA1。TimeStamp 签名的作用是防止重放攻击。该签名有效期为 60 秒。
Spas-SecurityToken	String	否	SecurityToken 需从 STS 临时凭证中获取。STS 临时凭证需从实例元数据 URL 中获取。详情请参考： <ul style="list-style-type: none"> <li>· 借助于实例 RAM 角色访问其他云产品</li> <li>· 通过 ECS 实例 RAM 角色访问 ACM</li> </ul>

返回参数

参数	类型	描述
configInfo	String	已变更配置的 Data ID、Group、命名空间信息，格式为 dataId^2group^2tenant^1 <ul style="list-style-type: none"> <li>· 配置的多个字段间分隔符：(Java) ^2 = Character.toString((char) 2) 或 (Shell) ^2=%02。</li> <li>· 配置间分隔符：(Java) ^1 = Character.toString((char) 1) 或 (Shell) ^1=%01</li> </ul>

错误码

错误码	错误信息	描述
400	Bad Request	客户端请求中的语法错误
403	Forbidden	没有权限
404	Not Found	客户端错误，未找到。
500	Internal Server Error	服务器内部错误

## 代码示例

### · 请求示例（Shell）

```
#!/bin/bash
## config param
dataId="com.alibaba.nacos.example.properties"
group="DEFAULT_GROUP"
namespace="04754ad1-4f67-4d67-b2bf-1f73a04a****"
accessKey="8c5cbb849ae04682ad9f455a96aa****"
secretKey="lw05T7vfPJU27FclPa+/CyIG****"
endpoint="acm.aliyun.com"
## config param end
## get serverIp from address server
serverIp=`curl $endpoint:8080/diamond-server/diamond -s | awk '{a[NR]
]=${0}END{ srand();i=int(rand()*NR+1);print a[i]}'`
## config sign
timestamp=`echo ${$(date +%s%N)}/10000000`'
signStr=$namespace+$group+$timestamp
signContent=`echo -n $signStr | openssl dgst -hmac $secretKey -sha1
-binary | base64`

echo "Listening..."
curl -X POST -H "Spas-AccessKey:$accessKey -H "timeStamp:$
timestamp -H "Spas-Signature:$signContent -H "longPullingTimeout:
30000" "http://"$serverIp":8080/diamond-server/config.co" -d "Probe
-Modify-Request=com.alibaba.nacos.example.properties%02DEFAULT_
GROUP%02f4da32aa4289aa861974f949b639****%0204754ad1-4f67-4d67-b2bf-
1f73a04a****%01" -v
```



#### 说明:

由于 shell 脚本在 Windows 系统下编辑容易编码错误，建议在 Linux 系统里新建 shell 脚本文件，再将此段代码复制到文件中。

### · 返回示例（Shell）

```
# 如果配置有变更
com.alibaba.nacos.example.properties%02DEFAULT_GROUP%02f4da32aa
4289aa861974f949b639****%0204754ad1-4f67-4d67-b2bf-1f73a04a****%01

# 如果配置无变更，则返回空串
```

更多信息

[API 概览](#)

# 6 syncUpdateAll

使用 syncUpdateAll 接口发布 ACM 配置。

请求类型

POST

请求 URL

/diamond-server/basestone.do

请求参数

参数	类型	是否必需	描述
tenant	String	是	租户信息，对应 ACM 的命名空间 ID。
dataId	String	是	配置的 ID
group	String	是	配置的分组
content	String	是	配置的内容

Header 参数

参数	类型	是否必需	描述
Spas-AccessKey	String	是	在 ACM 控制台上的命名空间详情对话框内可获取 AccessKey。
timeStamp	String	是	以毫秒为单位的请求时间。
Spas-Signature	String	是	使用 SecretKey 对 “Tenant +TimeStamp” 签名 (SpasSigner.sign(tenant+ timeStamp, secretKey))，签名算法为 HmacSHA1。TimeStamp 签名的作用是防止重放攻击。该签名有效期为 60 秒。
Spas-SecurityToken	String	否	SecurityToken 需从 STS 临时凭证中获取。STS 临时凭证需从实例元数据 URL 中获取。详情请参考： · 借助于实例 RAM 角色访问其他云产品 · 通过 ECS 实例 RAM 角色访问 ACM

## 返回参数

参数	类型	描述
-	Boolean	是否成功

## 错误码

错误码	错误信息	描述
400	Bad Request	客户端请求中的语法错误
403	Forbidden	没有权限
404	Not Found	客户端错误，未找到。
500	Internal Server Error	服务器内部错误

## 代码示例

### · 请求示例（Shell）

```
#!/bin/bash
## config param
dataId="com.alibaba.nacos.example.properties2"
group="DEFAULT_GROUP"
namespace="04754ad1-4f67-4d67-b2bf-1f73a04a****"
accessKey="8c5cbb849ae04682ad9f455a96aa****"
secretKey="lw05T7vfPJU27FclPa+/CyIG****"
endpoint="acm.aliyun.com"
## config param end
## get serverIp from address server
serverIp=`curl $endpoint:8080/diamond-server/diamond -s | awk '{a[NR] = $0}END{ srand(); i = int(rand() * NR + 1); print a[i]}'`
## config sign
timestamp=`echo ${$(date +%s%N)}/1000000`'
signStr=$namespace+$group+$timestamp
signContent=`echo -n $signStr | openssl dgst -hmac $secretKey -sha1
-binary | base64`
## request to publish a config
curl -X POST -H "Spas-AccessKey:$accessKey -H "timeStamp:$
timestamp -H "Spas-Signature:$signContent "http://"$serverIp":8080
/diamond-server/basestone.do?method=syncUpdateAll" -d "dataId="$
dataId"&group="$group"&tenant="$namespace"&content="key1=value1 -v
```



#### 说明:

由于 shell 脚本在 Windows 系统下编辑容易编码错误，建议在 Linux 系统里新建 shell 脚本文件，再将此段代码复制到文件中。

- **返回示例**

```
True
```

更多信息

[API 概览](#)

# 7 deleteAllDatums

---

使用 `deleteAllDatums` 接口删除 ACM 配置。

请求类型

**POST**

请求 URL

*/diamond-server/datum.do*

请求参数

参数	类型	是否必需	描述
tenant	String	是	租户信息，对应 ACM 的命名空间 ID。
dataId	String	是	配置的 ID
group	String	是	配置的分组

Header 参数

返回参数

参数	类型	是否必需	描述
Spas-AccessKey	String	是	在 ACM 控制台上的命名空间详情对话框内可获取 AccessKey。
timeStamp	String	是	以毫秒为单位的请求时间。
Spas-Signature	String	是	使用 SecretKey 对 “Tenant +TimeStamp” 签名 ( <code>SpasSigner.sign(tenant+ timeStamp, secretKey)</code> )，签名算法为 HmacSHA1。TimeStamp 签名的作用是防止重放攻击。该签名有效期为 60 秒。

错误码

错误码	错误信息	描述
400	Bad Request	客户端请求中的语法错误
403	Forbidden	没有权限
404	Not Found	客户端错误，未找到。

错误码	错误信息	描述
500	Internal Server Error	服务器内部错误

代码示例

· 请求示例（Shell）

```
#!/bin/bash
## config param
dataId="com.alibaba.nacos.example.properties2"
group="DEFAULT_GROUP"
namespace="04754ad1-4f67-4d67-b2bf-1f73a04a****"
accessKey="8c5cbb849ae04682ad9f455a96aa****"
secretKey="lw05T7vfPJU27FclPa+/CyIG****"
endpoint="acm.aliyun.com"
## config param end
## get serverIp from address server
serverIp=`curl $endpoint:8080/diamond-server/diamond -s | awk '{a[NR] = $0}END{ srand(); i=int(rand()*NR+1); print a[i]}'`
## config sign
timestamp=`echo ${$(date +%s%N)}/1000000` 
signStr=$group+$timestamp
signContent=`echo -n $signStr | openssl dgst -hmac $secretKey -sha1
-binary | base64` 
## request to delete a config
curl -X POST -H "Spas-AccessKey:$accessKey -H "timeStamp:$
timestamp -H "Spas-Signature:$signContent "http://"$serverIp":8080
/diamond-server/datum.do?method=deleteAllDatums" -d "dataId="$dataId
"&group="$group -v
```



说明：

由于 shell 脚本在 Windows 系统下编辑容易编码错误，建议在 Linux 系统里新建 shell 脚本文件，再将此段代码复制到文件中。

· 返回示例

True

更多信息

[API 概览](#)