

# API 网关

## 使用手册（调用API）

# 使用手册（调用API）

## 调用API

您可以通过API网关，调用由其他阿里云用户或者第三方服务商开放的API服务。API网关将为您提供一系列管理服务与支撑。

## 调用示例

您可以基于API网关提供的SDK，通过编写代码的方式调用API。也可以自行编辑HTTP请求调用API。API的请求结构如下：

//假如，域名：a13db7999e494a90819cce500130034d.com

//假如，Path：/web/cloudapi/mapping/service

//假如，Query内容是：a=name，b=12

//则请求的url如下

```
http://a13db7999e494a90819cce500130034d.com/web/cloudapi/mapping/service ? a=name&b=12
```

//请求的Method

```
POST HttpMethod : POST
```

//Headers中需要包含签名信息和部分参数

//详细加密签名的计算传递方式，请查看文档--入口及协议。

```
X-Ca-Version: 1 //API版本
X-Ca-Signature-Headers: X-Ca-Version,X-Ca-Key,X-Ca-Stage,X-Ca-Timestamp //参与签名的Header
X-Ca-Key: 60028305 //AppKey
X-Ca-Stage: test //Stage
X-Ca-Timestamp: 1456905123049 //时间戳
X-Ca-Signature: UAaH/qteir4G9UK4YR+NWdyq+c1rjl0PvtO/C1Qo68U= //签名
```

//Http标准头

```
Host: a13db7999e494a90819cce500130034d.com //服务地址
Date: Wed 02 Mar 2016 07:52:02 GMT
```

```
User-Agent: Apache-HttpClient/4.1.2 (java 1.6)
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

//Body内容

```
Amount=11&InstanceId=ClientInstanceId&InstanceName=ClientInstanceName
```

以上内容加上API的入参就构成了API请求。其中，在公测阶段，您需要在线下从提供API的服务方获取API的说明文档，获取服务地址、Path等详细说明。AppKey是您创建的APP的密钥，用于做身份验证。APP是您调用API需要创建的身份。详情将在后续内容中说明。

## 应用（APP）

您需要创建应用（APP）作为您调用API的身份，每个APP有一对AppKey和AppSecret密钥对，用于您请求时做加密签名，网关会对签名做验证。

1. 在API网关，您需要创建应用（APP）作为请求者的身份。APP创建时，系统会自动分配一对AppKey和AppSecret，AppKey用于标识您的身份，AppSecret是用于加密签名字符串和服务器端验证签名字符串的密钥。调用API时您需要将AppKey和AppSecret加密签名放入请求参数，API网关会通过对称加密来验证您的身份。详细加密签名的计算传递方式，请查看文档--入口及协议。
2. AppKey和AppSecret密钥对，具备该APP的全部权限，需要妥善保管。如果发生泄漏，您可以在API网关的控制台进行重置。
3. 您可以拥有多个APP，可以根据您的业务需求分别被授权不同的API。注意，API的授权对象是APP而不是阿里云用户账号。
4. 您可以在API网关控制台完成对APP的创建、修改、删除、查看详情、密钥管理、查看已授权等管理操作。

## 授权

授权，是指授予APP调用某个API的权限。您的APP需要获得API的授权才能调用该API。在公测阶段，由API服务方建立APP和API之间的权限关系。

1. 在公测阶段，由提供API的服务方来建立授权关系，您需要向其提供AppID或者您的阿里云邮箱账户，说明给予某一个APP进行授权。授权建立后，您就可以使用该APP调用API了。
2. 在公测阶段，您不具备建立或者删除授权的权限。您仅可以在控制台查看相应APP下已经被授权的API。如果您需要解除某个API的授权，请您联系API服务提供方。

## 加密签名

您调用API时，需要使用AppKey和AppSecret计算加密签名，完成身份验证。

1. 在API网关，使用APP作为您调用API的身份。APP创建时，系统会自动分配AppKey和AppSecret。服务器端会通过验证AppKey和AppSecret来完成请求者的身份验证。
2. 无论使用HTTP还是HTTPS协议提交请求，都需要在请求中包含签名信息。AppKey用于标识您的身份，AppSecret是用于加密签名字符串和服务器端验证签名字符串的密钥。详细加密签名的计算传递方式，请查看文档--入口及协议。

## 使用限制

- 每个账号下APP个数上限为10个，APP名称应为账号下唯一。
- 在公测阶段，您不具备授权或解除授权的权限，由API服务方完成相关操作。
- 您的请求需要包含签名信息，请参照文档--入口及协议

## 访问入口

### 域名

- 每个API服务都属于一个API分组，每个API分组有不同的域名。域名是由服务端绑定的独立域名，API网关通过域名来寻址定位API分组。
- 域名的格式为 `www.[独立域名].com/[Path]?[HTTPMethod]`，在公测阶段，API使用者需要在线下从API提供者处获取此域名。
- 云API网关通过域名定位到一个唯一的分组，通过Path+HTTPMethod确定该分组下唯一的API。
- 您需要在线下从API服务商处获取API文档，其中会包含域名、Path等必要参数信息。

## 系统级Header

- 【必选】X-Ca-Key：AppKey
- 【必选】X-Ca-Signature：签名字符串
- 【可选】X-Ca-Timestamp：API调用者传递时间戳，为时间转换为毫秒的值，也就是从1970年1月1日起至今的时间转换为毫秒，可选，默认15分钟内有效
- 【可选】X-Ca-Nonce：API调用者生成UUID，结合时间戳进行防重放，可选
- 【可选】Content-MD5：当请求Body非Form表单时，可以计算Body的MD5值传递给云网关进行Body MD5校验
- 【可选】X-Ca-Stage：请求API所属Stage，目前仅支持test和release，默认release。若您调用的API不在线上环境，请一定要指定该参数的值，否则会报URL错误。

## 签名校验

签名计算的详细demo（JAVA）请参照链接：<https://github.com/aliyun/api-gateway-demo-sign-java>。

签名计算方法步骤如下：

## 组织参与签名计算的字符串

```
String stringToSign=
HTTPMethod + "\n" +
Accept + "\n" +
Content-MD5 + "\n"
Content-Type + "\n" +
Date + "\n" +
Headers +
Url
```

### 说明

HTTPMethod为全大写

Accept、Content-MD5、Content-Type、Date如果为空也需要添加换行符"\n"，Headers如果为空不需要添加"\n"，不为空的Headers中包含了"\n"，详见下面组织Headers的描述

### Content-MD5

Content-MD5是指Body的MD5值，只有当Body非Form表单时才计算MD5，计算方式为：

```
String content-MD5 = Base64.encodeBase64(MD5(bodyStream.getBytes("UTF-8")));
```

bodyStream为字节数组

### Headers

Headers指所有参与签名计算的Header的Key、Value，这里要注意参与签名计算的Header是不包含X-Ca-Signature、X-Ca-Signature-Headers的。

Headers组织方法：

先对所有参与签名计算的Header的Key按照字典排序，然后按照如下方式拼接：

```
String headers =
HeaderKey1 + ":" + HeaderValue1 + "\n"+
HeaderKey2 + ":" + HeaderValue2 + "\n"+
...
HeaderKeyN + ":" + HeaderValueN + "\n"
```

### Url

Url指Path + Query + Body中Form参数，组织方法：

对Query+Form参数按照字典对Key进行排序后按照如下方法拼接，如果Query或Form参数为空，则Url = Path，不需要添加"?"

```
String url =  
Path +  
"?" +  
Key1 + "=" + Value1 +  
&" + Key2 + "=" + Value2 +  
...  
&" + KeyN + "=" + ValueN
```

注意这里Query或Form参数的Value可能有多个，多个的时候只取第一个Value参与签名计算

## 计算签名

```
Mac hmacSha256 = Mac.getInstance("HmacSHA256");  
byte[] keyBytes = secret.getBytes("UTF-8");  
hmacSha256.init(new SecretKeySpec(keyBytes, 0, keyBytes.length, "HmacSHA256"));  
String sign = new String(Base64.encodeBase64(Sha256.doFinal(stringToSign.getBytes("UTF-8")), "UTF-8"));
```

secret为APP对应的密钥

## 传递签名

将计算的签名放到Request的Header中，Key为：X-Ca-Signature

将所有参与加签的Header的Key使用英文逗号分割放到Request的Header中，不区分顺序，Key为：X-Ca-Signature-Headers

签名计算的详细demo请参照链接：<https://github.com/aliyun/api-gateway-demo-sign-java>。