ALIBABA CLOUD

Alibaba Cloud

访问控制

API 参考（STS）

文档版本：20211112

阿里云

法律声明

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

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

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

目录

1.什么是STS 05
2.请求结构 07
3.接入地址 08
4.公共参数 11
5.签名机制 13
6.返回结果 16
7.获取AccessKey 17
8.操作接口 19
8.1. AssumeRole 19
8.2. GetCallerIdentity 23
8.3. AssumeRoleWithSAML 24
8.4. AssumeRoleWithOIDC 29

1.什么是STS

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

应用场景

- **使用RAM用户扮演角色时获取STS Token**

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

  通常用于跨账号访问场景和临时授权场景。更多信息，请参见使用RAM角色、跨阿里云账号的资源授权和移动应用使用临时安全令牌访问阿里云。

- **角色SSO时获取STS Token**

  进行角色SSO时，通过调用AssumeRoleWithSAML接口，以获取某个RAM角色的STS Token，从而使用STS Token进行单点登录（SSO登录）。更多信息，请参见SAML角色SSO概览。

产品优势

- 使用STS Token，减少长期访问密钥（Accesskey）泄露的风险。
- STS Token具有时效性，可以自定义有效期，到期后将自动失效，无需定期轮换。
- 可以为STS Token绑定自定义权限策略，提供更加灵活和精细的云资源授权。

基本概念

概念	说明
RAM用户	RAM用户是RAM的一种实体身份类型，有确定的身份ID和身份凭证，它通常与某个确定的人或应用程序一一对应。 • 一个阿里云账号下可以创建多个RAM用户，对应企业内的员工、系统或应用程序。 • RAM用户不拥有资源，不能独立计量计费，由所属阿里云账号统一控制和付费。 • RAM用户归属于阿里云账号，只能在所属阿里云账号的空间下可见，而不是独立的阿里云账号。 • RAM用户必须在获得阿里云账号的授权后才能登录控制台或使用API操作阿里云账号下的资源。 更多信息，请参见RAM用户概览和创建RAM用户。

概念	说明
RAM角色	RAM角色是一种虚拟用户，可以被授予一组权限策略。与RAM用户不同，RAM角色没有确定的登录密码或访问密钥，它需要被一个可信的实体用户（RAM用户、阿里云服务或身份提供商）扮演。扮演成功后实体用户将获得RAM角色的临时身份凭证，即安全令牌（STS Token），使用安全令牌就能以RAM角色身份访问被授权的资源。 根据不同的可信实体，RAM角色分为以下三类： • **阿里云账号**：允许RAM用户所扮演的角色。扮演角色的RAM用户可以属于自己的阿里云账号，也可以属于其他阿里云账号。此类角色主要用来解决跨账号访问和临时授权问题。 • **阿里云服务**：允许云服务所扮演的角色。此类角色主要用于授权云服务代理您进行资源操作。 • **身份提供商**：允许可信身份提供商下的用户所扮演的角色。此类角色主要用于实现与阿里云的单点登录（SSO）。 更多信息，请参见RAM角色概览、创建可信实体为阿里云账号的RAM角色、创建可信实体为阿里云服务的RAM角色和创建可信实体为身份提供商的RAM角色。
角色ARN	角色ARN是角色的全局资源描述符，用来指定具体角色。ARN遵循阿里云ARN的命名规范。例如，某个阿里云账号下的devops角色的ARN为：`acs:ram::123456789012****:role/samplerole`。创建角色后，单击角色名后，可在**基本信息**页查看其ARN。
可信实体	RAM角色的可信实体是指可以扮演RAM角色的实体用户身份。创建RAM角色时必须指定可信实体，RAM角色只能被可信实体扮演。可信实体可以是阿里云账号、受信的阿里云服务或身份提供商。
权限策略	权限策略是用语法结构描述的一组权限的集合，可以精确地描述被授权的资源集、操作集以及授权条件。权限策略是描述权限集的一种简单语言规范。一个RAM角色可以绑定一组权限策略，没有绑定权限策略的RAM角色可以存在，但不能访问资源。
扮演角色	扮演角色是实体用户获取角色身份的安全令牌的方法。一个实体用户调用STS API AssumeRole可以获得角色的安全令牌，使用安全令牌可以访问云服务API。

支持STS的云服务

关于支持STS的云服务详情，请参见支持STS的云服务。

2.请求结构

本文介绍了STS的接入地址、通信协议、HTTPS请求方法和请求参数等请求结构相关的信息。

接入地址

STS的接入地址请参见接入地址。

通信协议

为了保证通信的安全性，STS服务仅支持使用HTTPS安全通道发送请求。

HTTPS请求方法

STS服务支持HTTPS GET/POST方法发送请求。

> **说明** GET请求最大不得超过4 KB， POST请求最大不得超过10 MB。

请求参数

每个请求都需要指定以下信息：

- 要执行的操作：Action参数。
- 每个操作接口都需要包含的公共请求参数。更多信息，请参见公共参数。
- 操作接口所特有的请求参数。

字符编码

请求及返回结果都使用UTF-8字符集进行编码。

3.接入地址

本文介绍了STS服务的所有接入地址，每个地址的功能都相同，请尽量在同地域进行调用。

以下表格分别罗列了各个地域的接入地址，其中接入地址一列包含如下信息：

- 公网：互联网访问地址。
- VPC：可在同地域内VPC中访问，无需开放公网访问权限。

亚太

地域名称	地域ID	接入地址
中国内地	不涉及	公网：sts.aliyuncs.com 说明 在中国内地，您可以直接使用该统一公网接入地址，也可以使用所在地域的接入地址。
华东1（杭州）	cn-hangzhou	• 公网：sts.cn-hangzhou.aliyuncs.com • VPC：sts-vpc.cn-hangzhou.aliyuncs.com
华东2（上海）	cn-shanghai	• 公网：sts.cn-shanghai.aliyuncs.com • VPC：sts-vpc.cn-shanghai.aliyuncs.com
华南1（深圳）	cn-shenzhen	• 公网：sts.cn-shenzhen.aliyuncs.com • VPC： sts-vpc.cn-shenzhen.aliyuncs.com
华南2（河源）	cn-heyuan	公网：sts.cn-heyuan.aliyuncs.com
华北1（青岛）	cn-qingdao	公网：sts.cn-qingdao.aliyuncs.com
华北2（北京）	cn-beijing	• 公网：sts.cn-beijing.aliyuncs.com • VPC：sts-vpc.cn-beijing.aliyuncs.com

地域名称	地域ID	接入地址
华北3（张家口）	cn-zhangjiakou	• 公网：sts.cn-zhangjiakou.aliyuncs.com • VPC：sts-vpc.cn-zhangjiakou.aliyuncs.com
华北5（呼和浩特）	cn-huhehaote	• 公网：sts.cn-huhehaote.aliyuncs.com • VPC：sts-vpc.cn-huhehaote.aliyuncs.com
华北6（乌兰察布）	cn-wulanchabu	• 公网：sts.cn-wulanchabu.aliyuncs.com • VPC：sts-vpc.cn-wulanchabu.aliyuncs.com
西南1（成都）	cn-chengdu	• 公网：sts.cn-chengdu.aliyuncs.com • VPC：sts-vpc.cn-chengdu.aliyuncs.com
中国（香港）	cn-hongkong	• 公网：sts.cn-hongkong.aliyuncs.com • VPC：sts-vpc.cn-hongkong.aliyuncs.com
新加坡	ap-southeast-1	• 公网：sts.ap-southeast-1.aliyuncs.com • VPC：sts-vpc.ap-southeast-1.aliyuncs.com
澳大利亚（悉尼）	ap-southeast-2	• 公网：sts.ap-southeast-2.aliyuncs.com • VPC：sts-vpc.ap-southeast-2.aliyuncs.com
马来西亚（吉隆坡）	ap-southeast-3	• 公网：sts.ap-southeast-3.aliyuncs.com • VPC：sts-vpc.ap-southeast-3.aliyuncs.com
印度尼西亚（雅加达）	ap-southeast-5	• 公网：sts.ap-southeast-5.aliyuncs.com • VPC：sts-vpc.ap-southeast-5.aliyuncs.com

地域名称	地域ID	接入地址
日本（东京）	ap-northeast-1	• 公网：sts.ap-northeast-1.aliyuncs.com • VPC：sts-vpc.ap-northeast-1.aliyuncs.com

欧洲与美洲

地域名称	地域ID	接入地址
美国（硅谷）	us-west-1	• 公网：sts.us-west-1.aliyuncs.com • VPC：sts-vpc.us-west-1.aliyuncs.com
美国（弗吉尼亚）	us-east-1	• 公网：sts.us-east-1.aliyuncs.com • VPC：sts-vpc.us-east-1.aliyuncs.com
德国（法兰克福）	eu-central-1	• 公网：sts.eu-central-1.aliyuncs.com • VPC：sts-vpc.eu-central-1.aliyuncs.com
英国（伦敦）	eu-west-1	• 公网：sts.eu-west-1.aliyuncs.com • VPC：sts-vpc.eu-west-1.aliyuncs.com

中东与印度

地域名称	地域ID	接入地址
印度（孟买）	ap-south-1	• 公网：sts.ap-south-1.aliyuncs.com • VPC：sts-vpc.ap-south-1.aliyuncs.com
阿联酋（迪拜）	me-east-1	公网：sts.me-east-1.aliyuncs.com

4.公共参数

公共参数分为公共请求参数和公共返回参数。

公共请求参数

名称	类型	是否必须	描述
Format	String	否	返回消息的格式。取值： • JSON（默认值） • XML
Version	String	是	API版本号，使用YYYY-MM-DD日期格式。取值：2015-04-01。
Signature	String	是	消息签名。更多信息，请参见签名机制。
SignatureMethod	String	是	签名方式。取值：HMAC-SHA1。
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。
SignatureVersion	String	是	签名算法版本。取值：1.0。
AccessKeyId	String	是	访问密钥ID。
Timestamp	String	是	请求的时间戳，为日期格式。使用UTC时间并按照ISO8601标准，格式为：YYYY-MM-DDThh:mm:ssZ。 例如：北京时间2013年01月10日20点00分00秒，表示为2013-01-10T12:00:00Z。

公共请求参数示例如下：

```
https://sts.aliyuncs.com?Action=AssumeRole
&Format=xml
&Version=2015-04-01
&Signature=Pc5WB8gokVn0xfeu%2FZV%2BiNM1dg****
&SignatureMethod=HMAC-SHA1
&SignatureNonce=1521552885****
&SignatureVersion=1.0
&AccessKeyId=LTAI4GENiH2u8MVj7Khh****
&Timestamp=2015-06-01T12:00:00Z
```

公共返回参数

API返回结果采用统一格式，调用成功返回的数据格式有XML和JSON两种，可以在发送请求时指定返回的数据格式，默认为JSON格式。每次接口调用，无论成功与否，系统都会返回一个唯一识别码RequestId。

- 返回 `2xx` HTTP状态码表示调用成功。
- 返回 `4xx` 或 `5xx` HTTP状态码表示调用失败。

公共返回参数示例如下：

- XML格式

```
<?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格式

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

5.签名机制

为保证API的安全调用，在调用API时阿里云会对每个API请求通过签名（Signature）进行身份验证。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。

概述

RPC API要按以下格式在API请求的Query中增加签名（Signature）。

```
https://Endpoint/?SignatureVersion=1.0&SignatureMethod=HMAC-SHA1&Signature=CT9X0VtwR86fNWSnsc6v8YGOjuE%3D&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf
```

其中：

- SignatureMethod：签名方式，目前支持HMAC-SHA1。
- SignatureVersion：签名算法版本，目前版本是1.0。
- SignatureNonce：唯一随机数，用于防止网络重放攻击。用户在不同请求间要使用不同的随机数值，建议使用通用唯一识别码UUID（Universally Unique Identifier）。
- Signature：使用AccessKey Secret对请求进行对称加密后生成的签名。

签名算法遵循RFC 2104 HMAC-SHA1规范，使用AccessKey Secret对编码、排序后的整个请求串计算HMAC值作为签名。签名的元素是请求自身的一些参数，由于每个API请求内容不同，所以签名的结果也不尽相同。可参考本文的操作步骤，计算签名值。

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

步骤一：构造待签名字符串

1. 使用请求参数构造规范化的请求字符串（Canonicalized Query String）。
   i. 按照参数名称的字典顺序对请求中所有的请求参数（包括公共请求参数和接口的自定义参数，但不包括公共请求参数中的Signature参数）进行排序。

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

ii. 对排序之后的请求参数的名称和值分别用UTF-8字符集进行URL编码。编码规则请参考下表。

字符	编码方式
A~Z、a~z和0~9以及“-”、“_”、“.”和“~”	不编码。
其它字符	编码成 `%XY` 的格式，其中 `XY` 是字符对应ASCII码的16进制表示。例如英文的双引号（"）对应的编码为 `%22` 。
扩展的UTF-8字符	编码成 `%XY%ZA…` 的格式。

英文空格	编码成 `%20` ，而不是加号（+）。
	该编码方式和一般采用的 `application/x-www-form-urlencoded` MIME格式编码算法（例如Java标准库中的 `java.net.URLEncoder` 的实现）存在区别。编码时可以先用标准库的方式进行编码，然后把编码后的字符串中的加号（+）替换成 `%20` ，星号（*）替换成 `%2A` ， `%7E` 替换回波浪号（~），即可得到上述规则描述的编码字符串。本算法可以用下面的 `percentEncode` 方法来实现：
	`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;`
	`}`

iii. 将编码后的参数名称和值用英文等号（=）进行连接。

iv. 将等号连接得到的参数组合按步骤i排好的顺序依次使用“&”符号连接，即得到规范化请求字符串。

2. 将构造的规范化字符串按照下面的规则构造成待签名的字符串。

```
StringToSign=
    HTTPMethod + “&” +
    percentEncode(“/”) + ”&” +
    percentEncode(CanonicalizedQueryString)
```

其中：

- HTTPMethod 是提交请求用的HTTP方法，例如GET。
- percentEncode(“/”) 是按照URL编码规则对字符“/”进行编码得到的值，即%2F。
- percentEncode(CanonicalizedQueryString) 是对构造的规范化请求字符串按URL编码规则编码后得到

的字符串。

步骤二：计算签名值

1. 按照RFC2104的定义，计算待签名字符串（StringToSign）的HMAC值。

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

2. 按照Base64编码规则把上面的HMAC值编码成字符串，即得到签名值（Signature）。
3. 将得到的签名值作为Signature参数添加到请求参数中。

> **说明** 得到的签名值在作为最后的请求参数值提交时要和其它参数一样，按照RFC3986的规则进行URL编码。

示例

下文以AssumeRole为例，介绍签名的一个具体示例及结果。

签名前的请求URL为：

```
https://sts.aliyuncs.com/?SignatureVersion=1.0&Format=JSON&Timestamp=2015-09-01T05%3A57%3A34Z&RoleArn=acs%3Aram%3A%3A1234567890123%3Arole%2Ffirstrole&RoleSessionName=client&AccessKeyId=testid&SignatureMethod=HMAC-SHA1&Version=2015-04-01&Action=AssumeRole&SignatureNonce=571f8fb8-506e-11e5-8e12-b8e8563dc8d2
```

对应的 `StringToSign` 为：

```
GET&%2F&AccessKeyId%3Dtestid%26Action%3DAssumeRole%26Format%3DJSON%26RoleArn%3Dacs%253Aram%253A%253A1234567890123%253Arole%252Ffirstrole%26RoleSessionName%3Dclient%26SignatureMethod%3DHMAC-SHA1%26SignatureNonce%3D571f8fb8-506e-11e5-8e12-b8e8563dc8d2%26SignatureVersion%3D1.0%26Timestamp%3D2015-09-01T05%253A57%253A34Z%26Version%3D2015-04-01
```

例如：AccessKey ID为testid，AccessKey Secret为testsecret，则用于计算HMAC的key为testsecret&。

计算得到的签名值为：`gNI7b0AyKZHxDgjBGPDgJ1Ce3L4=` 。

签名后的请求URL为：

```
https://sts.aliyuncs.com/?SignatureVersion=1.0&Format=JSON&Timestamp=2015-09-01T05%3A57%3A34Z&RoleArn=acs%3Aram%3A%3A1234567890123%3Arole%2Ffirstrole&RoleSessionName=client&AccessKeyId=testid&SignatureMethod=HMAC-SHA1&Version=2015-04-01&Signature=gNI7b0AyKZHxDgjBGPDgJ1Ce3L4%3D&Action=AssumeRole&SignatureNonce=571f8fb8-506e-11e5-8e12-b8e8563dc8d2
```

6.返回结果

调用STS API后返回数据采用统一格式，返回结果主要有XML和JSON两种格式，默认为XML格式。本文档中的返回示例为了便于您查看，做了格式化处理，实际返回结果是没有进行换行、缩进等处理的。

成功结果

调用STS API后，如果返回的HTTP状态码为 `2xx` ，代表调用成功。

- 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 */
}
```

错误结果

调用STS API后，如果返回的HTTP状态码为 `4xx` 或 `5xx` ，代表调用失败，系统将不会返回结果数据。此时，返回的消息体中包含：具体的错误代码、错误信息、全局唯一的请求ID `RequestId` 以及本次请求访问的站点ID `HostId` 。您可以通过错误码定位问题。

- XML示例

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <RequestId>8906582E-6722-409A-A6C4-0E7863B733A5</RequestId>
  <HostId>sts.aliyuncs.com</HostId>
  <Code>InvalidParameter</Code>
  <Message>The specified parameter "Action or Version" is not valid.</Message>
</Error>
```

- JSON示例

```
{
  "RequestId": "7463B73D-35CC-4D19-A010-6B8D65D242EF",
  "HostId": "sts.aliyuncs.com",
  "Code": "InvalidParameter",
  "Message": "The specified parameter \"Action or Version\" is not valid."
}
```

7.获取AccessKey

您可以为阿里云账号（主账号）和RAM用户创建一个访问密钥（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。

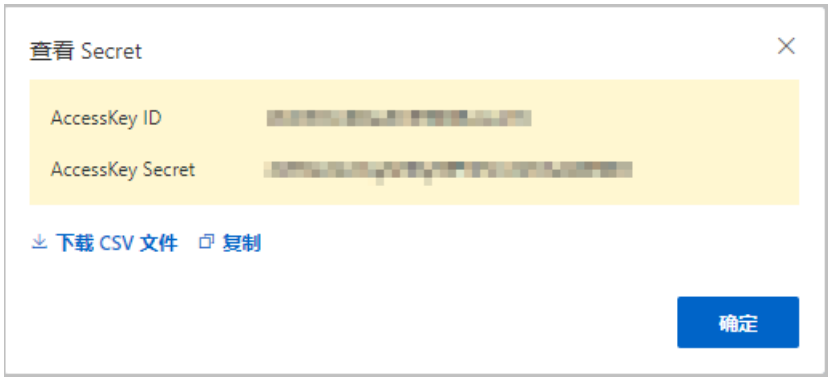
- AccessKey ID：用于标识用户。
- AccessKey Secret：用于验证用户的密钥。AccessKey Secret必须保密。

> **警告** 阿里云账号AccessKey泄露会威胁您所有资源的安全。建议您使用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. 使用阿里云账号登录控制台。
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**对话框，选择使用阿里云账号AccessKey或RAM用户AccessKey。

- 使用阿里云账号AccessKey
  a. 单击**继续使用AccessKey**。
  b. 在**AccessKey**页面，单击**创建AccessKey**。
  c. 在**查看Secret**对话框，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。



- 使用RAM用户AccessKey
  a. 单击**开始使用子用户AccessKey**。

b. 系统自动跳转到RAM控制台的**用户**页面，找到需要获取AccessKey的RAM用户。

> **说明** 如果没有RAM用户，请先创建RAM用户，详情请参见创建RAM用户。

c. 单击用户登录名称。

d. 在**认证管理**页签下的**用户AccessKey**区域，单击**创建AccessKey**。

e. 在**查看Secret**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。

> **说明**
>
> - RAM用户的AccessKey Secret只在创建时显示，不提供查询，请妥善保管。
> - 若AccessKey泄露或丢失，则需要创建新的AccessKey，最多允许为每个RAM用户创建2个AccessKey。

8.操作接口

8.1. AssumeRole

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

使用限制

- 本接口仅限具有STS角色扮演权限的RAM用户使用。没有权限时会报错，报错信息如下：

  ```
  You are not authorized to do this action. You should be authorized by RAM.
  ```

  问题原因和解决方法如下：

  - 该RAM用户缺少允许STS扮演角色的权限策略：请为该RAM用户添加系统策略（AliyunSTSAssumeRoleAccess）或自定义策略。更多信息，请参见为RAM用户授权和能否指定RAM用户具体可以扮演哪个RAM角色？。
  - RAM角色的信任策略不包含您正在使用的RAM用户，即RAM角色不允许该RAM用户扮演：请为RAM角色添加允许该RAM用户扮演的信任策略。更多信息，请参见修改RAM角色的信任策略。

- 本接口调用次数限制：每分钟最多调用6000次，且一个阿里云账号及该账号下的RAM用户、RAM角色共用这6000次。

  当请求量超过6000次时，超出部分会报错，报错信息如下：

  ```
  Request was denied due to user flow control.
  ```

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	AssumeRole	要执行的操作。取值：AssumeRole
RoleArn	String	是	acs:ram::1234567890 12****:role/adminrole	要扮演的RAM角色ARN。 该角色是可信实体为阿里云账号类型的RAM角色。更多信息，请参见创建可信实体为阿里云账号的RAM角色或CreateRole。 格式：`acs:ram::$accountID:role/$roleName`。 **说明** 您可以通过RAM控制台或API查看角色ARN。具体如下： • RAM控制台：请参见如何查看RAM角色的ARN？。 • API：请参见ListRoles或GetRole。

名称	类型	是否必选	示例值	描述
RoleSessionName	String	是	alice	角色会话名称。 该参数为用户自定义参数。通常设置为调用该API的用户身份，例如：用户名。在操作审计日志中，即使是同一个RAM角色执行的操作，也可以根据不同的RoleSessionName来区分实际操作者，以实现用户级别的访问审计。 长度为2~64个字符，可包含英文字母、数字、半角句号（.）、at（@）、短划线（-）和下划线（_）。
Policy	String	否	{"Statement": [{"Action": ["*"],"Effect": "Allow","Resource": ["*"]}],"Version":"1"}	为STS Token额外添加的一个权限策略，进一步限制STS Token的权限。具体如下： • 如果指定该权限策略，则STS Token最终的权限策略取RAM角色权限策略与该权限策略的交集。 • 如果不指定该权限策略，则STS Token最终的权限策略取RAM角色的权限策略。 长度为1~1024个字符。
DurationSeconds	Long	否	3600	过期时间。单位：秒。 过期时间最小值为900秒，最大值为要扮演角色的MaxSessionDuration时间。默认值为3600秒。 **说明** 您可以通过CreateRole或UpdateRole接口设置角色最大会话时间MaxSessionDuration。详情请参见CreateRole或UpdateRole。

返回数据

名称	类型	示例值	描述
RequestId	String	6894B13B-6D71-4EF5-88FA-F32781734A7F	请求ID。
Credentials			访问凭证。
└AccessKeyId	String	STS.L4aBSCSJVMuKg5U1****	访问密钥ID。

名称	类型	示例值	描述
└AccessKeySecret	String	wyLTSmsyPGP1ohvvw8xYgB29dlGI8KMiH2pK****	访问密钥。
└SecurityToken	String	********	安全令牌。
└Expiration	String	2015-04-09T11:52:19Z	失效时间。
AssumedRoleUser			角色扮演时的临时身份。
└Arn	String	acs:ram::1234567889012****:role/adminrole/alice	临时身份的ARN。
└AssumedRoleId	String	34458433936495****:alice	临时身份的ID。

示例

请求示例

```
https://sts.aliyuncs.com/?Action=AssumeRole
&RoleArn=acs:ram::123456789012****:role/adminrole
&RoleSessionName=alice
&DurationSeconds=3600
&<公共请求参数>
```

> **说明** 关于公共请求参数的详情，请参见公共参数。

正常返回示例

`XML` 格式

```
<AssumeRoleResponse>
  <RequestId>6894B13B-6D71-4EF5-88FA-F32781734A7F</RequestId>
  <AssumedRoleUser>
    <Arn>acs:ram::123456789012****:role/adminrole/alice</arn>
    <AssumedRoleId>34458433936495****:alice</AssumedRoleId>
  </AssumedRoleUser>
  <Credentials>
    <AccessKeyId>STS.L4aBSCSJVMuKg5U1****</AccessKeyId>
    <AccessKeySecret>wyLTSmsyPGP1ohvvw8xYgB29dlGI8KMiH2pK****</AccessKeySecret>
    <SecurityToken>********</SecurityToken>
    <Expiration>2015-04-09T11:52:19Z</Expiration>
  </Credentials>
</AssumeRoleResponse>
```

JSON 格式

```
{
  "Credentials": {
    "AccessKeyId": "STS.L4aBSCSJVMuKg5U1****",
    "AccessKeySecret": "wyLTSmsyPGP1ohvvw8xYgB29dlGI8KMiH2pK****",
    "Expiration": "2015-04-09T11:52:19Z",
    "SecurityToken": "********"
  },
  "AssumedRoleUser": {
    "Arn": "acs:ram::123456789012****:role/adminrole/alice",
    "AssumedRoleId":"34458433936495****:alice"
    },
  "RequestId": "6894B13B-6D71-4EF5-88FA-F32781734A7F"
}
```

错误码

HttpCode	错误码	错误信息	描述
400	InvalidParameter	The parameter RoleArn is wrongly formed.	角色ARN格式错误。
400	InvalidParameter.RoleArn	The parameter RoleArn is wrongly formed.	角色ARN格式错误。
400	InvalidParameter.RoleSessionName	The parameter RoleSessionName is wrongly formed.	RoleSessionName格式错误，支持输入2~64个字符，请输入至少2个字符，允许输入英文字母、数字、半角句号（.）、at（@）、短划线（-）和下划线（_）。
400	InvalidParameter.DurationSeconds	The Min/Max value of DurationSeconds is 15min/1hr.	DurationSeconds参数设置错误，过期时间最小值为900秒，最大值为MaxSessionDuration设置的时间。
400	InvalidParameter.PolicyGrammar	The parameter Policy has not passed grammar check.	权限策略语法错误。
400	InvalidParameter.PolicySize	The size of Policy must be smaller than 1024 bytes.	权限策略长度超限，最大不超过1024字符。
403	NoPermission	You are not authorized to do this action. You should be authorized by RAM.	STS Token没有权限。解决方法请参见为什么使用STS时会报错。

HttpCode	错误码	错误信息	描述
404	EntityNotExist.Role	The specified Role not exists.	指定的RAM角色不存在。
500	InternalError	STS Server Internal Error happened.	服务器内部错误。

后续步骤

获得扮演RAM角色的STS Token后，RAM用户便可以使用STS Token，开始以RAM角色身份访问阿里云资源。

8.2. GetCallerIdentity

调用GetCallerIdentity接口获取当前调用者的身份信息。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetCallerIdentity	要执行的操作。取值：GetCallerIdentity。

返回数据

名称	类型	示例值	描述
RequestId	String	2C9BE469-4A35-44D5-9529-CAA280B11603	请求ID。
Arn	String	acs:ram::1968132000012****:user/admin	当前调用者的ARN。
AccountId	String	196813200012****	当前调用者所属阿里云账号ID。
UserId	String	216959339000****	• 如果当前调用者是RAM用户，则返回当前调用者的RAM用户UID。 • 如果当前调用者是阿里云账号，则返回当前调用者阿里云账号ID。
RoleId	String	33537620082992****	如果当前调用者是RAM角色，则返回当前调用者的RAM角色ID。
PrincipalId	String	28877424437521****	身份标识。
IdentityType	String	RAMUser	身份类型。

示例

请求示例

```
https://sts.aliyuncs.com/?Action=GetCallerIdentity
&<公共请求参数>
```

返回示例

`XML` 格式

```
<GetCallerIdentityResponse>
  <RequestId>2C9BE469-4A35-44D5-9529-CAA280B11603</RequestId>
  <AccountId>196813200012****</AccountId>
  <UserId>216959339000****</UserId>
  <IdentityType>RAMUser</IdentityType>
  <PrincipalId>28877424437521****</PrincipalId>
  <Arn>acs:ram::196813200012****:user/admin</Arn>
</GetCallerIdentityResponse>
```

`JSON` 格式

```
{
  "RequestId": "2C9BE469-4A35-44D5-9529-CAA280B11603",
  "AccountId": "196813200012****",
  "UserId": "216959339000****",
  "IdentityType": "RAMUser",
  "PrincipalId": "28877424437521****",
  "Arn": "acs:ram::196813200012****:user/admin"
}
```

错误码

无。

8.3. AssumeRoleWithSAML

进行角色SSO时，通过调用AssumeRoleWithSAML接口，可以获取一个扮演该角色的临时身份凭证（STS Token）。

请求参数

> **说明** 由于AssumeRoleWithSAML接口使用SAML断言进行身份认证，可以匿名访问，因此不需要提供公共参数中的Signature、SignatureMethod、SignatureVersion和AccessKeyId参数。

名称	类型	是否必选	示例值	描述
Action	String	是	AssumeRoleWithSAML	要执行的操作。取值：AssumeRoleWithSAML。

名称	类型	是否必选	示例值	描述
SAMLProviderArn	String	是	acs:ram::123456789012****:saml-provider/company1	RAM中创建的身份提供商的ARN。 格式：`acs:ram::<account ID>:saml-provider/<saml_provider_ID>`。 **说明** 您可以通过RAM控制台或API查看身份提供商的ARN。具体如下： • RAM控制台：请参见查看SAML身份提供商基本信息。 • API：请参见GetSAMLProvider或ListSAMLProviders。
RoleArn	String	是	acs:ram::123456789012****:role/adminrole	要扮演的角色的ARN。 该角色是可信实体为身份提供商的RAM角色。更多信息，请参见创建可信实体为身份提供商的RAM角色或CreateRole。 格式：`acs:ram::<accountID>:role/<roleName>`。 **说明** 您可以通过RAM控制台或API查看角色ARN。具体如下： • RAM控制台：请参见如何查看RAM角色的ARN？。 • API：请参见ListRoles或GetRole。
SAMLAssertion	String	是	<base64_encoded_saml_assertion>	Base64编码后的SAML断言。 长度为4~100000个字符。 **说明** 需要从IdP获取完整的SAMLResponse，不能是单独的SAMLAssertion字段。

名称	类型	是否必选	示例值	描述
Policy	String	否	<url_encoded_policy>	为STS Token额外添加的一个权限策略，进一步限制STS Token的权限。具体如下： • 如果指定该权限策略，则STS Token最终的权限取RAM角色权限策略与该权限策略的交集。 • 如果不指定该权限策略，则STS Token最终的权限就是RAM角色的权限。 长度为1~1024个字符。
DurationSeconds	Long	否	3600	过期时间，单位为秒。 过期时间最小值为900秒，最大值为MaxSessionDuration设置的时间。默认值为3600秒。 **说明** 您可以通过CreateRole或UpdateRole接口设置角色最大会话时间MaxSessionDuration。详情请参见CreateRole或UpdateRole。

返回数据

名称	类型	示例值	描述
RequestId	String	6894B13B-6D71-4EF5-88FA-F32781734A7F	请求ID。
Credentials			访问凭证。
└AccessKeyId	String	STS.L4aBSCSJVMuKg5U1****	访问密钥ID。
└AccessKeySecret	String	wyLTSmsyPGP1ohvvw8xYgB29dlGI8KMiH2pK****	访问密钥。
└SecurityToken	String	********	安全令牌。
└Expiration	String	2015-04-09T11:52:19Z	失效时间。
AssumedRoleUser			角色扮演临时身份。

名称	类型	示例值	描述
└Arn	String	acs:sts::123456789012****:assumed-role/AdminRole/alice	临时身份的ARN。
└AssumedRoleId	String	34458433936495****:alice	临时身份的ID。
SAMLAssertionInfo			SAML断言中的部分信息。
└SubjectType	String	persistent	SAML断言中 `NameID` 的格式。当前缀为 `urn:oasis:names:tc:SAML:2.0:nameid-format:` 时，前缀会被移除。例如： `persistent/transient` 。
└Subject	String	alice@example.com	SAML断言中 `Subject - NameID` 字段的值。
└Recipient	String	https://signin.aliyun.com/saml-role/SSO	SAML断言中 `Subiect - SubiectConfirmation - SubiectConfirmationData` 字段中 `Recipient` 属性的值。
└Issuer	String	http://example.com/adfs/services/trust	SAML断言中 `Issuer` 字段的值。

示例

请求示例

```
https://sts.aliyuncs.com/?Action=AssumeRoleWithSAML
&Format=xml
&Version=2015-04-01
&SignatureNonce=1521552885****
&Timestamp=2021-06-01T12:00:00Z
&RoleArn=acs:ram::123456789012****:role/adminrole
&SAMLProviderArn=acs:ram::123456789012****:saml-provider/company1
&SAMLAssertion=<base64_encoded_saml_assertion>
&Policy=<url_encoded_policy>
&DurationSeconds=3600
```

> **说明** 由于SAMLAssertion参数较长，可能会导致GET请求失败，请您使用POST方法发送此请求。

返回示例

`XML` 格式

```
<AssumeRoleResponse>
  <RequestId>6894B13B-6D71-4EF5-88FA-F32781734A7F</RequestId>
  <AssumedRoleUser>
    <arn>acs:sts::123456789012****:assumed-role/AdminRole/alice</arn>
    <AssumedRoleId>34458433936495****:alice</AssumedRoleId>
  </AssumedRoleUser>
  <Credentials>
    <AccessKeyId>STS.L4aBSCSJVMuKg5U1****</AccessKeyId>
    <AccessKeySecret>wyLTSmsyPGP1ohvvw8xYgB29dlGI8KMiH2pK****</AccessKeySecret>
    <SecurityToken>********</SecurityToken>
    <Expiration>2015-04-09T11:52:19Z</Expiration>
  </Credentials>
  <SAMLAssertionInfo>
    <SubjectType>persistent</SubjectType>
    <Subject>alice@example.com</Subject>
    <Recipient>https://signin.aliyun.com/saml-role/SSO</Recipient>
    <Issuer>http://example.com/adfs/services/trust</Issuer>
  </SAMLAssertionInfo>
</AssumeRoleResponse>
```

`JSON` 格式

```
{
  "Credentials": {
    "AccessKeyId": "STS.L4aBSCSJVMuKg5U1****",
    "AccessKeySecret": "wyLTSmsyPGP1ohvvw8xYgB29dlGI8KMiH2pK****",
    "Expiration": "2015-04-09T11:52:19Z",
    "SecurityToken": "********"
  },
  "AssumedRoleUser": {
    "arn": "acs:sts::1234567890123456:assumed-role/AdminRole/alice",
    "AssumedRoleId":"34458433936495****:alice"
    },
  "SAMLAssertionInfo": {
    "SubjectType": "persistent",
    "Subject": "alice@example.com",
    "Recipient": "https://signin.aliyun.com/saml-role/SSO",
    "Issuer": "http://example.com/adfs/services/trust"
  },
  "RequestId": "6894B13B-6D71-4EF5-88FA-F32781734A7F"
}
```

错误码

HttpCode	错误码	错误信息	描述
400	MissingParameter.SAMLAssertion	Parameter SAMLAssertion is required.	缺少SAMLAssertion参数。

HttpCode	错误码	错误信息	描述
400	MissingParameter.SAMLProviderArn	Parameter SAMLProviderArn is required.	缺少SAMLProviderArn参数。
400	MissingParameter.RoleArn	Parameter RoleArn is required.	缺少RoleArn参数。
400	InvalidParameter.PolicyGrammar	Invalid Policy.	无效的权限策略。
400	InvalidParameter.PolicySize	The max size of policy string is 1024.	权限策略字符串长度超限，最大不超过1024字符。
400	InvalidParameter.RoleSessionName	The RoleSessionName is invalid.	角色会话名称无效。
400	InvalidParameter.DurationSeconds	The DurationSeconds is invalid.	DurationSeconds无效。
404	EntityNotExist.SAMLProvider	Can not find SAML provider.	找不到SAML身份提供商。
404	EntityNotExist.RoleArn	The specified Role does not exists.	指定的角色不存在。
401	AuthenticationFail.IDPMetadata.Invalid	The IdP Metadata of your SAML Provider is invalid.	SAML身份提供商的IdP元数据无效。
401	AuthenticationFail.SAMLAssertion.Expired	The SAML Assertion is expired.	SAML断言已过期。
401	AuthenticationFail.SAMLAssertion.Invalid	The SAML Assertion is invalid.	SAML断言无效。

8.4. AssumeRoleWithOIDC

进行OIDC角色SSO时，通过调用AssumeRoleWithOIDC接口，获取扮演RAM角色的临时身份凭证（STS Token）。

前提条件

- 确保已从外部身份提供商（IdP）获取到OIDC令牌（OIDC Token）。
- 确保已在RAM中创建了OIDC身份提供商。具体操作，请参见创建OIDC身份提供商或CreateOIDCProvider 。
- 确保已在RAM中创建了可信实体为OIDC身份提供商的RAM角色。具体操作，请参见创建可信实体为身份提供商的RAM角色或CreateRole 。

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	AssumeRoleWithOIDC	要执行的操作。取值：**AssumeRoleWithOIDC**。
OIDCProviderArn	String	是	acs:ram::113511544585****:oidc-provider/TestOidcIdp	OIDC身份提供商的ARN。 您可以通过RAM控制台或API查看OIDC身份提供商的ARN。具体如下： • RAM控制台：请参见查看OIDC身份提供商信息。 • API：请参见GetOIDCProvider或ListOIDCProviders 。
RoleArn	String	是	acs:ram::113511544585****:role/testoidc	需要扮演的RAM角色ARN。 您可以通过RAM控制台或API查看角色ARN。具体如下： • RAM控制台：请参见如何查看RAM角色的ARN。 • API：请参见ListRoles或GetRole 。
OIDCToken	String	是	eyJraWQiOiJKQzl3eHpyaHFKMGd0****	由外部IdP签发的OIDC令牌（OIDC Token）。 长度：4~20000个字符。 **说明** 需要输入原始OIDC Token，无需Base64解码。
Policy	String	否	{"Statement":[{"Action":["*"],"Effect":"Allow","Resource":["*"]}],"Version":"1"}	为STS Token额外添加的一个权限策略，进一步限制STS Token的权限。具体如下： • 如果指定该权限策略，则STS Token最终的权限取RAM角色权限策略与该权限策略的交集。 • 如果不指定该权限策略，则STS Token最终的权限就是RAM角色的权限。 长度：1~1024个字符。

名称	类型	是否必选	示例值	描述
DurationSeconds	Long	否	3600	会话过期时间。单位：秒。 默认值：3600。最小值：900。最大值：`MaxSessionDuration` 设置的时间。 **说明** 关于设置角色最大会话时间 `MaxSessionDuration` 的具体操作，请参见CreateRole或UpdateRole。
RoleSessionName	String	否	TestOidcAssumedRoleSession	角色会话名称。 该参数为用户自定义参数。通常设置为调用该API的用户身份，例如：用户名。在操作审计日志中，即使是同一个RAM角色执行的操作，也可以根据不同的RoleSessionName来区分实际操作者，以实现用户级别的访问审计。 格式：包含英文字母、数字、半角句号（.）、at（@）、短划线（-）和下划线（_）。 长度：2~64个字符。

说明 由于AssumeRoleWithOIDC接口使用OIDC Token进行身份认证，可以匿名访问，因此不需要提供公共参数中的 `Signature` 、 `SignatureMethod` 、 `SignatureVersion` 和 `AccessKeyId` 参数。关于公共请求参数的详情，请参见公共参数。

返回数据

名称	类型	示例值	描述
RequestId	String	3D57EAD2-8723-1F26-B69C-F8707D8B565D	请求ID。
OIDCTokenInfo	Object		解析的OIDC Token信息。
Subject	String	KryrkIdjylZb7agUgCEf****	OIDC主体。 对应OIDC Token中的 `sub` 字段值。
Issuer	String	https://dev-xxxxxx.okta.com	OIDC颁发者URL。 对应OIDC Token中的 `iss` 字段值。

名称	类型	示例值	描述
ClientIds	String	496271242565057****	OIDC受众。多个之间用半角逗号（,）分隔。 对应OIDC Token中的 `aud` 字段值。
AssumedRoleUser	Object		角色扮演临时身份。
AssumedRoleId	String	33157794895460****	临时身份的ID。
Arn	String	acs:ram::113511544585****:role/testoidc/TestOidcAssumedRoleSession	临时身份的ARN。
Credentials	Object		临时访问凭证（STS Token）。
SecurityToken	String	CAIShwJ1q6Ft5B2yfSjIr5bSEsj4g7BihPWGWHz****	安全令牌。
Expiration	String	2021-10-20T04:27:09Z	失效时间。
AccessKeySecret	String	CVwjCkNzTMupZ8NbTCxCBRq3K16jtcWFTJAyBEv2****	访问密钥。
AccessKeyId	String	STS.NUgYrLnoC37mZZCNnAbez****	访问密钥ID。

示例

请求示例

```
https://sts.aliyuncs.com/?Action=AssumeRoleWithOIDC
&OIDCProviderArn=acs:ram::113511544585****:oidc-provider/TestOidcIdp
&RoleArn=acs:ram::113511544585****:role/testoidc
&OIDCToken=eyJraWQiOiJKQzl3eHpyaHFKMGd0****
&Policy={"Statement": [{"Action": ["*"],"Effect": "Allow","Resource": ["*"]}],"Version":"1"}
&DurationSeconds=3600
&RoleSessionName=TestOidcAssumedRoleSession
&公共请求参数
```

正常返回示例

`XML` 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<?xml version="1.0" encoding="UTF-8" ?>
<AssumeRoleWithOIDCResponse>
 <RequestId>3D57EAD2-8723-1F26-B69C-F8707D8B565D</RequestId>
 <OIDCTokenInfo>
  <Subject>KryrkIdjylZb7agUgCEf****</Subject>
  <Issuer>https://dev-xxxxxx.okta.com</Issuer>
  <ClientIds>496271242565057****</ClientIds>
 </OIDCTokenInfo>
 <AssumedRoleUser>
  <AssumedRoleId>33157794895460****</AssumedRoleId>
  <Arn>acs:ram::113511544585****:role/testoidc/TestOidcAssumedRoleSession</Arn>
 </AssumedRoleUser>
 <Credentials>
  <SecurityToken>CAIShwJ1q6Ft5B2yfSjIr5bSEsj4g7BihPWGWHz****</SecurityToken>
  <Expiration>2021-10-20T04:27:09Z</Expiration>
  <AccessKeySecret>CVwjCkNzTMupZ8NbTCxCBRq3K16jtcWFTJAyBEv2****</AccessKeySecret>
  <AccessKeyId>STS.NUgYrLnoC37mZZCNnAbez****</AccessKeyId>
 </Credentials>
</AssumeRoleWithOIDCResponse>
```

`JSON` 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
 "RequestId" : "3D57EAD2-8723-1F26-B69C-F8707D8B565D",
 "OIDCTokenInfo" : {
  "Subject" : "KryrkIdjylZb7agUgCEf****",
  "Issuer" : "https://dev-xxxxxx.okta.com",
  "ClientIds" : "496271242565057****"
 },
 "AssumedRoleUser" : {
  "AssumedRoleId" : "33157794895460****",
  "Arn" : "acs:ram::113511544585****:role/testoidc/TestOidcAssumedRoleSession"
 },
 "Credentials" : {
  "SecurityToken" : "CAIShwJ1q6Ft5B2yfSjIr5bSEsj4g7BihPWGWHz****",
  "Expiration" : "2021-10-20T04:27:09Z",
  "AccessKeySecret" : "CVwjCkNzTMupZ8NbTCxCBRq3K16jtcWFTJAyBEv2****",
  "AccessKeyId" : "STS.NUgYrLnoC37mZZCNnAbez****"
 }
}
```

错误码

访问错误中心查看更多错误码。