

阿里云 访问控制

API参考

文档版本：20180909

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
	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，恢复业务所需时间约10分钟。
	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。	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。	单击 确定。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid Instance_ID</code>
<code>[]</code> 或者 <code>[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>swich {stand slave}</code>

目录

法律声明.....	I
通用约定.....	I
1 API 参考 (RAM)	1
1.1 简介.....	1
1.1.1 RAM简介.....	1
1.1.2 术语表.....	2
1.2 API overview.....	2
1.3 调用方式.....	6
1.3.1 请求结构.....	6
1.3.2 公共参数.....	7
1.3.3 返回结果.....	9
1.3.4 签名.....	10
1.4 用户管理接口.....	12
1.4.1 CreateUser.....	12
1.4.2 GetUser.....	16
1.4.3 UpdateUser.....	18
1.4.4 DeleteUser.....	22
1.4.5 ListUsers.....	24
1.4.6 CreateLoginProfile.....	27
1.4.7 GetLoginProfile.....	29
1.4.8 DeleteLoginProfile.....	31
1.4.9 UpdateLoginProfile.....	33
1.4.10 CreateAccessKey.....	35
1.4.11 UpdateAccessKey.....	37
1.4.12 DeleteAccessKey.....	39
1.4.13 ListAccessKeys.....	41
1.4.14 CreateVirtualMFADevice.....	43
1.4.15 ListVirtualMFADevices.....	45
1.4.16 DeleteVirtualMFADevice.....	47
1.4.17 BindMFADevice.....	48
1.4.18 UnbindMFADevice.....	51
1.4.19 GetUserMFAInfo.....	52
1.4.20 ChangePassword.....	54
1.5 组管理接口.....	56
1.5.1 CreateGroup.....	56
1.5.2 GetGroup.....	58
1.5.3 UpdateGroup.....	60
1.5.4 ListGroups.....	63
1.5.5 DeleteGroup.....	65

1.5.6 AddUserToGroup.....	67
1.5.7 RemoveUserFromGroup.....	69
1.5.8 ListGroupsForUser.....	72
1.5.9 ListUsersForGroup.....	74
1.6 角色管理接口.....	77
1.6.1 CreateRole.....	77
1.6.2 GetRole.....	80
1.6.3 UpdateRole.....	82
1.6.4 ListRoles.....	85
1.6.5 DeleteRole.....	87
1.7 授权策略管理接口.....	88
1.7.1 CreatePolicy.....	88
1.7.2 GetPolicy.....	91
1.7.3 DeletePolicy.....	93
1.7.4 ListPolicies.....	95
1.7.5 CreatePolicyVersion.....	98
1.7.6 GetPolicyVersion.....	100
1.7.7 DeletePolicyVersion.....	102
1.7.8 ListPolicyVersions.....	104
1.7.9 SetDefaultPolicyVersion.....	106
1.7.10 AttachPolicyToUser.....	108
1.7.11 DetachPolicyFromUser.....	111
1.7.12 AttachPolicyToGroup.....	113
1.7.13 DetachPolicyFromGroup.....	116
1.7.14 AttachPolicyToRole.....	118
1.7.15 DetachPolicyFromRole.....	121
1.7.16 ListEntitiesForPolicy.....	123
1.7.17 ListPoliciesForUser.....	127
1.7.18 ListPoliciesForGroup.....	129
1.7.19 ListPoliciesForRole.....	131
1.8 安全设置接口.....	133
1.8.1 SetAccountAlias.....	133
1.8.2 GetAccountAlias.....	134
1.8.3 ClearAccountAlias.....	135
1.8.4 SetPasswordPolicy.....	136
1.8.5 GetPasswordPolicy.....	139
1.8.6 SetSecurityPreference.....	140
1.9 数据类型.....	142
1.9.1 User.....	142
1.9.2 LoginProfile.....	143
1.9.3 MFADevice.....	143
1.9.4 VirtualMFADevice.....	144
1.9.5 AccessKey.....	145
1.9.6 Group.....	145

1.9.7 Role.....	146
1.9.8 Policy.....	147
1.9.9 PolicyVersion.....	148
1.9.10 PasswordPolicy.....	148
1.9.11 SecurityPreference.....	149
1.10 错误代码表.....	150
1.11 关于RAM限制.....	160
2 API 参考 (STS)	161
2.1 简介.....	161
2.2 调用方式.....	161
2.2.1 请求结构.....	161
2.2.2 公共参数.....	162
2.2.3 返回结果.....	163
2.2.4 签名.....	165
2.3 操作接口.....	167
2.3.1 AssumeRole.....	167
2.3.2 GetCallerIdentity.....	169
2.4 数据类型.....	171
2.4.1 Credentials.....	171
2.4.2 AssumedRoleUser.....	171
2.5 附录.....	172
2.5.1 错误代码表.....	172
2.5.2 服务地址.....	173

1 API 参考 (RAM)

1.1 简介

1.1.1 RAM简介

什么是RAM？

RAM (Resource Access Management) 是阿里云提供的资源访问控制服务。通过RAM，您可以集中管理您的用户（比如员工、系统或应用程序），以及控制用户可以访问您名下哪些资源的权限。

RAM要解决的问题

如果您购买了云资源，您的组织里有多个用户需要使用这些云资源，这些用户只能共享使用您的云账号密钥。这里有两个问题：(1) 您的密钥由多人共享，泄露的风险很高；(2) 您无法控制特定用户能访问哪些资源的权限。

RAM解决问题的方法

RAM允许您在您的云账户下创建多个RAM用户，并给这些用户分配资源操作权限。RAM用户不允许拥有资源，没有独立的计量计费，这些用户由您的云账户统一控制和付费。您可以给每个RAM用户创建单独的登录密码或访问密钥，但在默认条件下，这些用户并没有任何操作权限。RAM提供基于访问策略的授权机制，支持您实现对RAM用户的细粒度授权。

RAM功能

RAM包括下列功能：

- 集中控制RAM用户及其密钥 —— 您可以管理每个用户及其访问密钥，为用户绑定/解绑多因素认证设备
- 集中控制RAM用户的访问权限 —— 您可以控制每个用户可以访问您名下哪些资源的操作权限
- 集中控制RAM用户的资源访问方式 —— 您可以确保用户必须使用安全信道（如SSL）、在指定时间、以及在指定的网络环境下请求访问特定的云服务
- 集中控制云资源 —— 您可以对用户创建的实例或数据进行集中控制。当用户离开您的组织时，这些实例或数据不会丢失
- 统一账单 —— 您的账户将收到包括所有用户的资源操作所发生的费用的单一账单

RAM访问点

RAM的默认访问点地址是: <https://ram.aliyuncs.com>，用户必须使用https接入访问点。

支持RAM的云产品

几乎所有的阿里云产品都会与RAM集成。目前支持哪些服务与RAM集成的信息，请参阅 [支持 RAM 的云服务](#)。

RAM服务费用

RAM服务是您的阿里云账户的一个功能，无需额外付费。您只需为您的RAM用户所使用的其它服务付费。

1.1.2 术语表

术语	中文	说明
Account	账户	这里是指一个云账户
User	用户	这里是指一个账户下的RAM用户
Group	群组、用户组	这里是指一个账户下的RAM用户组
Policy	授权策略	用来描述授权策略的一种描述语言
AK (Access Key)	访问密钥	访问密钥由AccessKeyID和AccessKeySecret组成，用于云服务API请求的身份认证
MFA (Multi-Factor Authentication)	多因素认证	您可以给账户或RAM用户绑定MFA设备来增强安全性

1.2 API overview

This page lists all of the available RAM APIs. For more OpenAPI resources, go to [API Explorer](#).

User management APIs

API	Description
CreateUser	Used to create a RAM user
GetUser	Used to obtain user details
UpdateUser	Used to update basic user information
DeleteUser	Used to delete a RAM user
ZH-CN_TP_12390.dita	Used to list all RAM users

API	Description
CreateLoginProfile	Used to enable web console logon for a RAM user
GetLoginProfile	Used to view the logon configuration of a RAM user
DeleteLoginProfile	Used to disable web console logon for a RAM user
ZH-CN_TP_12394.dita	Used to modify user logon configuration
CreateAccessKey	Used to create an access key for a RAM user
UpdateAccessKey	Used to change the access key status of a RAM user
DeleteAccessKey	Used to delete the access key of a RAM user
ListAccessKeys	Used to list the access keys of specified users
CreateVirtualMFADevice	Used to create a virtual MFA device
ListVirtualMFADevices	Used to list virtual MFA devices
DeleteVirtualMFADevice	Used to delete virtual MFA devices
BindMFADevice	Used to bind an MFA device
UnbindMFADevice	Used to unbind an MFA device
GetUserMFAInfo	Used to obtain the MFA device bound to a specified RAM user
ChangePassword	Used to change the password for a RAM user

Group management APIs

API	Description
CreateGroup	Used to create a user group
GetGroup	Used to obtain user group information
UpdateGroup	Used to update user group information
ListGroups	Used to list all user groups
DeleteGroup	Used to delete a specified user group
AddUserToGroup	Used to add RAM users to a specified user group
RemoveUserFromGroup	Used to remove a RAM user from a user group

API	Description
ListGroupForUser	Used to list the information about the group to which a specified RAM user is added
ListUsersForGroup	Used to list the RAM users included in a specified user group

Role management APIs

API	Description
CreateRole	Used to create a role
GetRole	Used to obtain role information
UpdateRole	Used to update role information
ListRoles	Used to list roles
DeleteRole	Used to delete a specified role

Authorization policy management APIs

API	Description
CreatePolicy	Used to create an authorization policy
GetPolicy	Used to obtain information about a specified authorization policy
DeletePolicy	Used to delete a specified authorization policy
ListPolicies	Used to list authorization policies
CreatePolicyVersion	Used to create a new authorization policy version
GetPolicyVersion	Used to obtain the version information about an authorization policy
DeletePolicyVersion	Used to delete a version of a specified authorization policy
ListPolicyVersions	Used to list all versions of an authorization policy
SetDefaultPolicyVersion	Used to set the default version of an authorization policy
AttachPolicyToUser	Used to add an authorization policy to a specified user

API	Description
<i>DetachPolicyFromUser</i>	Used to remove an authorization policy from a user
<i>AttachPolicyToGroup</i>	Used to add an authorization policy to a specified user group
<i>DetachPolicyFromGroup</i>	Used to remove an authorization policy from a user group
<i>AttachPolicyToRole</i>	Used to add an authorization policy to a specified role
<i>DetachPolicyFromRole</i>	Used to remove an authorization policy from a role
<i>ListEntitiesForPolicy</i>	Used to list the entities under an authorization policy
<i>ListPoliciesForUser</i>	Used to list the authorization policies assigned to a specified user
<i>ListPoliciesForGroup</i>	Used to list the authorization policies assigned to a specified user group
<i>ListPoliciesForRole</i>	Used to list the authorization policies assigned to a specified role

Security setting APIs

API	Description
<i>SetAccountAlias</i>	Used to set a cloud account alias
<i>GetAccountAlias</i>	Used to view a cloud account alias
<i>ClearAccountAlias</i>	Used to remove a cloud account alias
<i>SetPasswordPolicy</i>	Used to set the password policy for a user, including the password strength
<i>GetPasswordPolicy</i>	Used to obtain the password policy of a user, including the password strength
<i>SetSecurityPreference</i>	Used to set the global security preferences

Data types

Type	Description
<i>User</i>	User information

Type	Description
LoginProfile	User logon configuration
MFADevice	MFA device
VirtualMFADevice	Virtual MFA device
AccessKey	Access key
Group	Group information
Role	Role
Policy	Authorization policy
PolicyVersion	Authorization policy version
PasswordPolicy	Password policy
SecurityPreference	Security preference Download PDF

1.3 调用方式

1.3.1 请求结构

服务地址

RAM服务的API接入地址为

```
https://ram.aliyuncs.com
```

通信协议

为了保证通信的安全性，RAM服务仅支持使用HTTPS安全通道发送请求。

HTTP请求方法

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

请求参数

需要在每次请求中使用参数Action指名要执行的操作，同时需要加入请求的公共参数及接口业务参数。

字符编码

请求及返回结果都使用UTF-8字符集进行编码。

1.3.2 公共参数

Format

- 名称: Format
- 类型: String
- 必须: 否
- 说明: 返回值的类型，支持JSON与XML，默认为XML。

Version

- 名称: Version
- 类型: String
- 必须: 是
- 说明: API版本号，为日期形式：YYYY-MM-DD，本版本对应为2015-05-01。

AccessKeyId

- 名称: AccessKeyId
- 类型: String
- 必须: 是
- 说明: 访问密钥ID

Signature

- 名称: Signature
- 类型: String
- 必须: 是
- 说明: 消息签名

SignatureMethod

- 名称: SignatureMethod
- 类型: String
- 必须: 是
- 说明: 签名方式，目前仅支持HMAC-SHA1。

SignatureVersion

- 名称: SignatureVersion

- 类型: String
- 必须: 是
- 说明: 签名算法版本，目前版本是1.0

SignatureNonce

- 名称: SignatureNonce
- 类型: String
- 必须: 是
- 说明: 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。

Timestamp

- 名称: Timestamp
- 类型: String
- 必须: 是
- 说明: 请求的时间戳。日期格式按照ISO8601标准表示，并需要使用UTC时间。格式为：

```
YYYY-MM-DDThh:mm:ssZ
```

例如，2013-01-10T12:00:00Z (为北京时间2013年1月10日20点0分0秒)

请求示例

```
https://ram.aliyuncs.com/  
  ?Format=xml  
  &Version=2015-05-01  
  &Signature=Pc5WB8gokVn0xfeu%2FZV%2BiNM1dgI%3D  
  &SignatureMethod=HMAC-SHA1  
  &SignatureNonce=15215528852396  
  &SignatureVersion=1.0  
  &AccessKeyId=key-test  
  &Timestamp=2012-06-01T12:00:00Z  
  ...
```

公共返回参数

用户发送的每次接口调用请求，无论成功与否，系统都会返回一个唯一识别码RequestId给用户。
此参数用于识别每一次请求。

示例

- **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 */
}
```

1.3.3 返回结果

调用API服务后返回数据采用统一格式，返回的HTTP状态码为2xx，代表调用成功；返回4xx或5xx的HTTP状态码代表调用失败。调用成功返回的数据格式主要有XML和JSON两种，外部系统可以在请求时传入参数来制定返回的数据格式，默认为XML格式。本文档中的返回示例为了便于用户查看，做了格式化处理，实际返回结果是没有进行换行、缩进等处理的。

成功结果

- **XML示例**

XML返回结果包括请求是否成功信息和具体的业务数据。示例如下：

```
<?xml version="1.0" encoding="utf-8"?>
<!--结果的根结点-->
<接口名称+Response>
  <!--返回请求标签-->
  <RequestId>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

调用接口出错后，将不会返回结果数据。调用方可根据[错误代码表](#)来定位错误原因。

当调用出错时，HTTP请求将返回4xx或5xx的HTTP状态码。返回的消息体中是具体的错误代码及错误信息。另外还包含一个全局唯一的请求ID：RequestId和一个您该次请求访问的站点ID：HostId。

在调用方找不到错误原因时，可以联系阿里云客服，并提供该HostId和RequestId，以便我们尽快帮您解决问题。

- **XML示例**

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

1.3.4 签名

RAM服务会对每个访问的请求进行身份验证，所以无论使用HTTP还是HTTPS协议提交请求，都需要在请求中包含签名（Signature）信息。RAM通过使用**Access Key ID**和**Access Key Secret**进行对称加密的方法来验证请求的发送者身份。**Access Key ID**和**Access Key Secret**由阿里云官方颁发给访问者（可以通过阿里云官方网站申请和管理），其中**Access Key ID**用于标识访问者的身份；**Access Key Secret**是用于加密签名字符串和服务器端验证签名字符串的密钥，必须严格保密，只有阿里云和用户知道。

签名步骤

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



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）。

b. 对每个请求参数的名称和值进行编码。名称和值要使用UTF-8字符集进行URL编码，URL编码的编码规则是：

- 对于字符 A-Z、a-z、0-9以及字符“-”、“_”、“.”、“~”不编码；
- 对于其他字符编码成“%XY”的格式，其中XY是字符对应ASCII码的16进制表示。比如英文的双引号 (") 对应的编码就是%22
- 需要说明的是英文空格 () 要被编码是%20，而不是加号 (+)。



说明：

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

c. 对编码后的参数名称和值使用英文等号 (=) 进行连接。

d. 再把英文等号连接得到的字符串按参数名称的字典顺序依次使用&符号连接，即得到规范化请求字符串。

2. 使用上一步构造的规范化字符串按照下面的规则构造用于计算签名的字符串：

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

其中HTTPMethod是提交请求用的HTTP方法，比GET。percentEncode("/")是按照1.b中描述的URL编码规则对字符"/"进行编码得到的值，即"%2F"。

percentEncode(CanonicalizedQueryString)是对第1步中构造的规范化请求字符串按1.b中描述的URL编码规则编码后得到的字符串。

3. 按照[RFC2104](#)的定义，使用上面的用于签名的字符串计算签名HMAC值。注意：计算签名时使用的Key就是用户持有的Access Key Secret并加上一个"&"字符(ASCII:38)，使用的哈希算法是SHA1。

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

5. 将得到的签名值作为Signature参数添加到请求参数中，即完成对请求签名的过程。注意：得到的签名值在作为最后的请求参数值提交给RAM服务器的时候，要和其他参数一样，按照RFC3986的规则进行URL编码）。

示例

以CreateUser为例，签名前的请求URL为：

```
https://ram.aliyuncs.com/?UserName=test&SignatureVersion=1.0&Format=JSON&Timestamp=2015-08-18T03%3A15%3A45Z&AccessKeyId=testid&SignatureMethod=HMAC-SHA1&Version=2015-05-01&Action=CreateUser&SignatureNonce=6a6e0ca6-4557-11e5-86a2-b8e8563dc8d2
```

对应的StringToSign是：

```
GET&%2F&AccessKeyId%3Dtestid%26Action%3DCreateUser%26Format%3DJSON%26SignatureMethod%3DHMAC-SHA1%26SignatureNonce%3D6a6e0ca6-4557-11e5-86a2-b8e8563dc8d2%26SignatureVersion%3D1.0%26Timestamp%3D2015-08-18T03%253A15%253A45Z%26UserName%3Dtest%26Version%3D2015-05-01
```

假如使用的Access Key Id是“testid”，Access Key Secret是“testsecret”，用于计算HMAC的Key就是“testsecret&”，则计算得到的签名值是：

```
kRA2cnpJVacIhDMzXnoNZG9tDCI%3D
```

签名后的请求URL为（注意增加了Signature参数）：

```
https://ram.aliyuncs.com/?UserName=test&SignatureVersion=1.0&Format=JSON&Timestamp=2015-08-18T03%3A15%3A45Z&AccessKeyId=testid&SignatureMethod=HMAC-SHA1&Version=2015-05-01&Signature=kRA2cnpJVacIhDMzXnoNZG9tDCI%3D&Action=CreateUser&SignatureNonce=6a6e0ca6-4557-11e5-86a2-b8e8563dc8d2
```

1.4 用户管理接口

1.4.1 CreateUser

Interface description

Creates an RAM user.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Operation interface, required. The parameter value is “CreateUser”.

UserName

- Type: String

- Required: Yes
- Description: User name. It consists of a maximum of 64 characters.
- Format:

```
^[a-zA-Z0-9\.\@\-\_]+$
```

DisplayName

- Type: String
- Required: No
- Description: Display name. It consists of a maximum of 12 characters.
- Format:

```
^[a-zA-Z0-9\.\@\-\u4e00-\u9fa5]+$
```

MobilePhone

- Type: String
- Required: No
- Description: An RAM user's mobile number.
- Format: International area code-number such as 86-18600008888

E-mail

- Type: String
- Required: No
- Description: An RAM user's email address.

Comments

- Type: String
- Required: No
- Description: Remark information. It consists of a maximum of 128 characters.

Required permissions

Action

```
ram:CreateUser
```

Resource

```
ACS: Ram: *: $ {accountant}: User /*
```

返回参数

User

- 类型 : [User](#) Type
- Description: user information

Error messages

InvalidParameter.UserName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP status: 400
- Error Message : The parameter - "UserName" beyond the length limit.

InvalidParameter.DisplayName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "DisplayName" contains invalid chars.

InvalidParameter.DisplayName.Length

- HTTP Status : 400
- Error Message : The parameter - "DisplayName" beyond the length limit.

InvalidParameter.Comments.Length

- HTTP Status : 400
- Error Message : The parameter - "Comments" beyond the length limit.

InvalidParameter.MobilePhone.Format

- HTTP Status : 400
- Error Message : The format of the parameter - "MobilePhone" is incorrect.

InvalidParameter.Email.Format

- HTTP Status : 400
- Error Message : The format of the parameter - "Email" is incorrect.

EntityAlreadyExists.User

- HTTP Status : 409
- Error Message : The user does already EXIST.

LimitExceeded.User

- HTTP Status : 409
- Error Message : The count of users beyond the current limits.

Operation examples**Request example****Note:**

To facilitate reading, parameters are not encoded in the following request examples.

```
https://ram.aliyuncs.com/?Action=CreateUser
&UserName=zhangqiang
&DisplayName=zhangqiang
&MobilePhone=86-18688888888
&Email=zhangqiang@example.com
&Comments=This is a cloud computing engineer.
&<Public request parameters>
```

Response example

- XML format

```
<CreateUserResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <User>
    <UserId>1227489245380721</UserId>
    <UserName>zhangqiang</UserName>
    <DisplayName>zhangqiang</DisplayName>
    <MobilePhone>86-18600008888</MobilePhone>
    <Email>zhangqiang@example.com</Email>
    <Comments>This is a cloud computing engineer.</Comments>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
  </User>
```

```
</CreateUserResponse>
```

- JSON format

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "User": {
    "UserId": "1227489245380721",
    "UserName": "zhangqiang",
    "DisplayName": "zhangqiang",
    "MobilePhone": "86-18600008888",
    "Email": "maid ",
    "Comments": "This is a cloud computing engineer.",
    "CreateDate": "2015-01-23T12:33:18Z"
  }
}
```

1.4.2 GetUser

接口描述

获取用户的详细信息

请求参数

Action

- 名称：Action
- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：GetUser

UserName

- 名称：UserName
- 类型：String
- 必须：是
- 描述：指定用户名。例:zhangqiang
- 格式：

```
^[a-zA-Z0-9\.\@\-\_]+$
```

返回参数

User

- 类型：[User](#)

- 描述：用户信息

需要的权限

Action

```
ram:GetUser
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=GetUser
&UserName=zhangqiang
&<公共请求参数>
```

返回示例

- XML格式

```
<GetUserResponse>
  <RequestId>2D69A58F-345C-4FDE-88E4-BF5189484043</RequestId>
  <User>
    <UserId>1227489245380721</UserId>
    <UserName>zhangqiang</UserName>
    <DisplayName>张强</DisplayName>
    <MobilePhone>86-18600008888</MobilePhone>
    <Email>zhangqiang@example.com</Email>
    <Comments>这是一位云计算工程师</Comments>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
```

```
<UpdateDate>2015-02-11T03:15:21Z</UpdateDate>
<LastLoginDate>2015-01-23T12:33:18Z</LastLoginDate>
</User>
<UserId>1227489245380721</UserId>
</GetUserResponse>
```

- JSON格式

```
{
  "RequestId": "2D69A58F-345C-4FDE-88E4-BF5189484043",
  "User": {
    "UserId": "1227489245380721",
    "UserName": "zhangqiang",
    "DisplayName": "张强",
    "MobilePhone": "86-18600008888",
    "Email": "zhangqiang@example.com",
    "Comments": "这是一位云计算工程师",
    "CreateDate": "2015-01-23T12:33:18Z",
    "UpdateDate": "2015-02-11T03:15:21Z",
    "LastLoginDate": "2015-01-23T12:33:18Z"
  }
}
```

1.4.3 UpdateUser

接口描述

更新用户的基本信息

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：UpdateUser

UserName

- 类型：String
- 必须：是
- 描述：指定用户名
- 格式：

```
^[a-zA-Z0-9\.\@\-\_]+$
```

NewUserName

- 类型：String

- 必须：否
- 描述：指定新用户名
- 格式：

```
^[a-zA-Z0-9\.\@\-\_]+$
```

NewDisplayName

- 类型：String
- 必须：否
- 描述：指定新显示名称

NewMobilePhone

- 类型：String
- 必须：否
- 描述：指定RAM用户新手机号
- 格式：国际区号-号码；例如：86-18600008888

NewEmail

- 类型：String
- 必须：否
- 描述：指定RAM用户的新邮箱

NewComments

- 类型：String
- 必须：否
- 描述：指定新备注, 最大长度128个字符。

返回参数

User

- 类型：[User](#)
- 描述：用户信息

需要的权限

Action

```
ram:UpdateUser
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

InvalidParameter.NewUserName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "NewUserName" contains invalid chars.

InvalidParameter.NewUserName.Length

- HTTP Status : 400
- Error Message : The parameter - "NewUserName" beyond the length limit.

InvalidParameter.NewDisplayName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "NewDisplayName" contains invalid chars.

InvalidParameter.NewDisplayName.Length

- HTTP Status : 400
- Error Message : The parameter - "NewDisplayName" beyond the length limit.

InvalidParameter.NewComments.Length

- HTTP Status : 400
- Error Message : The parameter - "NewComments" beyond the length limit.

InvalidParameter.NewMobilePhone.Format

- HTTP Status : 400
- Error Message : The format of the parameter - "NewMobilePhone" is incorrect.

InvalidParameter.NewEmail.Format

- HTTP Status : 400
- Error Message : The format of the parameter - "NewEmail" is incorrect.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

EntityAlreadyExists.User

- HTTP Status : 409
- Error Message : The user does already EXIST.

操作示例**请求示例**

```
https://ram.aliyuncs.com/?Action=UpdateUser
&UserName=zhangqiang
&NewUserName=xiaoqiang
&NewMobilePhone=86-18600008888
&NewEmail=zhangqiang@example.com
&<公共请求参数>
```

返回示例

- XML格式

```
<CreateUserResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <User>
    <UserId>1227489245380721</UserId>
    <UserName>xiaoqiang</UserName>
    <DisplayName>张强</DisplayName>
    <MobilePhone>86-18600008888</MobilePhone>
    <Email>zhangqiang@example.com</Email>
    <Comments>这是一位云计算工程师</Comments>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
    <UpdateDate>2015-02-11T03:15:21Z</UpdateDate>
  </User>
```

```
</CreateUserResponse>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "User": {
    "UserId": "1227489245380721",
    "UserName": "xiaoqiang",
    "DisplayName": "张强",
    "MobilePhone": "86-18600008888",
    "Email": "zhangqiang@example.com",
    "Comments": "这是一位云计算工程师",
    "CreateDate": "2015-01-23T12:33:18Z",
    "UpdateDate": "2015-02-11T03:15:21Z"
  }
}
```

1.4.4 DeleteUser

接口描述

删除一个RAM用户

删除用户前，需要保证用户不拥有任何授权策略(Policy)，并且不属于任何用户组。

请求参数

Action

- 名称：Action
- 类型：String
- 必须：是
- 描述：系统规定参数，取值：DeleteUser

UserName

- 名称：UserName
- 类型：String
- 必须：是
- 描述：指定用户名。例:zhangqiang
- 格式：

```
^[a-zA-Z0-9\.\@\-\_]+$
```

返回参数

只有公共返回参数，详见[公共参数](#)

需要的权限

Action

```
ram:DeleteUser
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

DeleteConflict.User.Group

- HTTP Status : 409
- Error Message : The user CAN NOT be in any group while deleting the user.

DeleteConflict.User.AccessKey

- HTTP Status : 409
- Error Message : The user CAN NOT has any access key while deleting the user.

DeleteConflict.User.LoginProfile

- HTTP Status : 409
- Error Message : The user CAN NOT has any login profile while deleting the user.

DeleteConflict.User.MFADevice

- HTTP Status : 409
- Error Message : The user CAN NOT has any mfa device while deleting the user.

DeleteConflict.User.Policy

- HTTP Status : 409
- Error Message : The user CAN NOT has any attached policy while deleting the user.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=DeleteUser
&UserName=zhangqiang
&<公共请求参数>
```

返回示例

- XML格式

```
<DeleteUserResponse>
  <RequestId>1C488B66-B819-4D14-8711-C4EAAA13AC01</RequestId>
</DeleteUserResponse>
```

- JSON格式

```
{
  "RequestId": "1C488B66-B819-4D14-8711-C4EAAA13AC01"
}
```

1.4.5 ListUsers

接口描述

列出所有RAM用户

请求参数

Action

- 名称 : Action
- 类型 : String
- 必须 : 是
- 描述 : 系统规定参数，取值：ListUsers

Marker

- 名称 : Marker
- 类型 : String
- 必须 : 否

- 描述：当请求的返回结果被截断时，可以使用Marker获取从当前截断位置之后的内容

MaxItems

- 名称：MaxItems
- 类型：Integer
- 必须：否
- 取值范围：[1 - 100]
- 默认值：100
- 描述：指定返回结果的条数，当返回结果达到MaxItems限制被截断时，返回参数IsTruncated将等于true

返回参数

IsTruncated

- 类型：Boolean
- 描述：请求返回结果是否被截断

Marker

- 类型：String
- 描述：当IsTruncated为true时才有此字段，当返回true时，需要继续调用此接口，并且使用Marker获取截断后的内容

Users

- 类型：[User](#)
- 描述：用户信息集合

需要的权限

Action

```
ram:ListUsers
```

Resource

```
acs:ram:*:${AccountId}:user/*
```

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=ListUsers
```

&<公共请求参数>

返回示例

- XML格式

```
<ListUsersResponse>
  <RequestId>4B450CA1-36E8-4AA2-8461-86B42BF4CC4E</RequestId>
  <Users>
    <User>
      <UserId>1227489245380721</UserId>
      <UserName>zhangqiang</UserName>
      <DisplayName>张强</DisplayName>
      <MobilePhone>86-18600008888</MobilePhone>
      <Email>zhangqiang@example.com</Email>
      <Comments>这是一位云计算工程师</Comments>
      <CreateDate>2015-01-23T12:33:18Z</CreateDate>
      <UpdateDate>2015-01-23T12:33:18Z</UpdateDate>
    </User>
    <User>
      <UserId>1406498224724456</UserId>
      <UserName>lili</UserName>
      <DisplayName>李丽</DisplayName>
      <MobilePhone>86-18600009999</MobilePhone>
      <Email>lili@example.com</Email>
      <Comments>权限管理员</Comments>
      <CreateDate>2015-02-18T17:22:08Z</CreateDate>
      <UpdateDate>2015-02-18T17:22:08Z</UpdateDate>
    </User>
  </Users>
  <IsTruncated>true</IsTruncated>
  <Marker>EXAMPLE</Marker>
</ListUsersResponse>
```

- JSON格式

```
{
  "RequestId" : "4B450CA1-36E8-4AA2-8461-86B42BF4CC4E",
  "IsTruncated": true,
  "Marker": "EXAMPLE",
  "Users" : {
    "User" : [
      {
        "UserId" : "1227489245380721",
        "UserName" : "zhangqiang",
        "DisplayName": "张强",
        "MobilePhone": "86-18600008888",
        "Email": "zhangqiang@example.com",
        "Comments": "这是一位云计算工程师",
        "CreateDate" : "2015-01-23T12:33:18Z",
        "UpdateDate" : "2015-01-23T12:33:18Z"
      },
      {
        "UserId" : "1406498224724456",
        "UserName" : "lili",
        "DisplayName": "李丽",
        "MobilePhone": "86-18600009999",
        "Email": "lili@example.com",
```

```
        "Comments": "权限管理员",
        "CreateDate" : "2015-02-18T17:22:08Z",
        "UpdateDate" : "2015-02-18T17:22:08Z"
    }
  ]
}
```

1.4.6 CreateLoginProfile

接口描述

为一个RAM子用户启用Web控制台登录

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：CreateLoginProfile

UserName

- 类型：String
- 必须：是
- 描述：指定用户名

Password

- 类型：String
- 必须：是
- 描述：指定密码，密码必须符合密码强度要求。关于密码强度设置接口，请参考[GetPasswordPolicy](#)。

PasswordResetRequired

- 类型：Boolean
- 必须：否
- 默认：False
- 描述：指定用户在登录时是否需要修改密码；

MFABindRequired

- 类型：Boolean

- 必须：否
- 默认：False
- 描述：指定用户在下次登录时是否必须绑定多因素认证器

返回参数

LoginProfile

- 类型：[LoginProfile](#)
- 描述：登陆配置信息

需要的权限

Action

```
ram:CreateLoginProfile
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status：400
- Error Message：The parameter - “UserName” contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status：400
- Error Message：The parameter - “UserName” beyond the length limit.

InvalidParameter.Password.TooWeak

- HTTP Status：400
- Error Message：The parameter - “Password” is not compliant with the password policy.

EntityNotExist.User

- HTTP Status：404
- Error Message：The user does not exist.

EntityAlreadyExists.User.LoginProfile

- HTTP Status：409

- Error Message : The user login profile does already EXIST.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=CreateLoginProfile
&UserName=zhangqiang
&Password=mypassword
&PasswordResetRequired=true
&<公共请求参数>
```

返回示例

- XML格式

```
<CreateLoginProfile>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <LoginProfile>
    <UserName>zhangqiang</UserName>
    <PasswordResetRequired>true</PasswordResetRequired>
    <MFABindRequired>true</MFABindRequired>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
  </LoginProfile>
</CreateLoginProfile>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "LoginProfile": {
    "UserName": "zhangqiang",
    "PasswordResetRequired": true,
    "MFABindRequired": true,
    "CreateDate": "2015-01-23T12:33:18Z"
  }
}
```

1.4.7 GetLoginProfile

接口描述

查看一个RAM用户的登录配置

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：GetLoginProfile

UserName

- 类型：String
- 必须：是
- 描述：指定用户名

返回参数

LoginProfile

- 类型：[LoginProfile](#)
- 描述：登陆配置信息

需要的权限

Action

```
ram:GetLoginProfile
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status：400
- Error Message：The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status：400
- Error Message：The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status：404
- Error Message：The user does not exist.

EntityNotExist.User.LoginProfile

- HTTP Status：404
- Error Message：The user login profile does not exist.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=GetLoginProfile
&UserName=zhangqiang
&<公共请求参数>
```

返回示例

- XML格式

```
<GetLoginProfile>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <LoginProfile>
    <UserName>zhangqiang</UserName>
    <PasswordResetRequired>true</PasswordResetRequired>
    <MFABindRequired>true</MFABindRequired>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
  </LoginProfile>
</GetLoginProfile>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "LoginProfile": {
    "UserName": "zhangqiang",
    "PasswordResetRequired": true,
    "MFABindRequired": true,
    "CreateDate": "2015-01-23T12:33:18Z"
  }
}
```

1.4.8 DeleteLoginProfile

接口描述

关闭指定子用户登录Web控制台的功能

请求参数

Action

- 类型：String
- 必须：是
- 描述：系统规定参数，取值：DeleteLoginProfile

UserName

- 类型：String
- 必须：是

- 描述：指定用户名。例:zhangqiang

返回参数

只有公共返回参数，详见[公共参数](#)。

需要的权限

Action

```
ram:DeleteLoginProfile
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

EntityNotExist.User.LoginProfile

- HTTP Status : 404
- Error Message : The user login profile does not exist.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=DeleteLoginProfile
&UserName=zhangqiang
&<公共请求参数>
```

返回示例

- XML格式

```
<DeleteLoginProfileResponse>
  <RequestId>1C488B66-B819-4D14-8711-C4EAAA13AC01</RequestId>
</DeleteLoginProfileResponse>
```

- JSON格式

```
{
  "RequestId": "1C488B66-B819-4D14-8711-C4EAAA13AC01"
}
```

1.4.9 UpdateLoginProfile

接口描述

修改用户的登陆配置

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：UpdateLoginProfile

UserName

- 类型：String
- 必须：是
- 描述：指定用户名

Password

- 类型：String
- 必须：否
- 描述：指定密码，密码必须符合主账号的密码强度要求；

PasswordResetRequired

- 类型：Boolean
- 必须：否
- 描述：指定用户在登录时是否需要修改密码；

MFABindRequired

- 类型：Boolean
- 必须：否
- 描述：指定用户在下次登录时是否必须绑定MFA

返回参数

LoginProfile

- 类型：[LoginProfile](#)
- 描述：登陆配置信息

需要的权限

Action

```
ram:UpdateLoginProfile
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status：400
- Error Message：The parameter - “UserName” contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status：400
- Error Message：The parameter - “UserName” beyond the length limit.

EntityNotExist.User

- HTTP Status：404
- Error Message：The user does not exist.

EntityNotExist.User.LoginProfile

- HTTP Status：404
- Error Message：The user login profile does not exist.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=UpdateLoginProfile
&UserName=zhangqiang
&Password=mypassword
&PasswordResetRequired=true
&<公共请求参数>
```

返回示例

- XML格式

```
<UpdateLoginProfile>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
</UpdateLoginProfile>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368"
}
```

1.4.10 CreateAccessKey

接口描述

为RAM子用户创建AccessKey

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：CreateAccessKey

UserName

- 类型：String
- 必须：否
- 描述：指定用户名，子账号调用此接口时，默认为自己创建AccessKey

返回参数

AccessKey

- 类型：[AccessKey](#)

- 描述：访问密钥

需要的权限

Action

```
ram:CreateAccessKey
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

LimitExceeded.User.AccessKey

- HTTP Status : 409
- Error Message : The access key count of the user access keys beyond the current limits.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=CreateAccessKey
&UserName=zhangqiang
&<公共请求参数>
```

返回示例

- XML格式

```
<CreateAccessKeyResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <AccessKey>
    <AccessKeyId>0wNEpMMLzy7szvai</AccessKeyId>
    <AccessKeySecret>PupkTg8jdmau1cXxYacgE736PJj4cA</AccessKeyS
ecret>
    <Status>Active</Status>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
  </AccessKey>
```

```
</CreateAccessKeyResponse>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "AccessKey": {
    "AccessKeyId": "0wNEpMMlzy7szvai",
    "AccessKeySecret": "PupkTg8jdmau1cXxYacgE736PJj4cA",
    "Status": "Active",
    "CreateDate": "2015-01-23T12:33:18Z"
  }
}
```

1.4.11 UpdateAccessKey

接口描述

变更RAM子用户AccessKey的状态

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：UpdateAccessKey

UserAccessKeyId

- 类型：String
- 必须：是
- 描述：指定要更新的AccessKeyId

Status

- 类型：String
- 必须：是
- 描述：AccessKey的状态, Active/Inactive

UserName

- 类型：String
- 必须：否
- 描述：指定用户名

返回参数

只有公共返回参数，详见[公共参数](#)。

需要的权限

Action

```
ram:UpdateAccessKey
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

EntityNotExist.User.AccessKey

- HTTP Status : 404
- Error Message : The user access key does not exist.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=UpdateAccessKey
&UserName=zhangqiang
&UserAccessKeyId=0wNEpMMLzy7szvai
&Status=Inactive
&<公共请求参数>
```

返回示例

- XML格式

```
<UpdateAccessKeyResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
</UpdateAccessKeyResponse>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368"
}
```

1.4.12 DeleteAccessKey

接口描述

删除RAM子用户的AccessKey

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：DeleteAccessKey

UserAccessKeyId

- 类型：String
- 必须：是
- 描述：指定要删除的AccessKeyId

UserName

- 类型：String
- 必须：否
- 描述：指定用户名

返回参数

只有公共返回参数，详见[公共参数](#)

需要的权限

Action

```
ram:DeleteAccessKey
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

EntityNotExist.User.AccessKey

- HTTP Status : 404
- Error Message : The user access key does not exist.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=DeleteAccessKey
&UserName=zhangqiang
&UserAccessKeyId=0wNEpMMLzy7szvai
&<公共请求参数>
```

返回示例

- XML格式

```
<DeleteAccessKeyResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
```

```
</DeleteAccessKeyResponse>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368"
}
```

1.4.13 ListAccessKeys

接口描述

列出指定用户的AccessKeys

请求参数

Action

- 类型：String
- 必须：是
- 描述：系统规定参数，取值：ListAccessKeys

UserName

- 类型：String
- 必须：否
- 描述：指定用户，子帐号访问时不提供此参数则表示列出自己的AccessKeys。

返回参数

AccessKeys

- 类型：[AccessKey](#) Array
- 描述：用户信息集合

需要的权限

Action

```
ram:ListAccessKeys
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

操作示例

请求示例

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

返回示例

- XML格式

```
<ListAccessKeysResponse>
  <RequestId>4B450CA1-36E8-4AA2-8461-86B42BF4CC4E</RequestId>
  <AccessKeys>
    <AccessKey>
      <AccessKeyId>0wNEpMMlzy7szvai</AccessKeyId>
      <Status>Active</Status>
      <CreateDate>2015-01-23T12:33:18Z</CreateDate>
    </AccessKey>
    <AccessKey>
      <AccessKeyId>WnIWUruvfDT37vQ</AccessKeyId>
      <Status>Inactive</Status>
      <CreateDate>2015-03-24T21:12:21Z</CreateDate>
    </AccessKey>
  </AccessKeys>
</ListAccessKeysResponse>
```

- JSON格式

```
{
  "RequestId" : "4B450CA1-36E8-4AA2-8461-86B42BF4CC4E",
  "AccessKeys":{
    "AccessKey":[
      {
        "AccessKeyId": "0wNEpMMlzy7szvai",
        "Status": "Active",
        "CreateDate": "2015-01-23T12:33:18Z"
      },
      {
        "AccessKeyId": "WnIWUruvfDT37vQ",
        "Status": "Inactive",
```

```
    "CreateDate": "2015-03-24T21:12:21Z"
  }
}
```

1.4.14 CreateVirtualMFADevice

接口描述

创建虚拟多因素认证设备

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：CreateVirtualMFADevice

VirtualMFADeviceName

- 类型：String
- 必须：是
- 描述：指定设备名称, 最大长度64个字符
- 限制：

```
[a-zA-Z0-9-]*
```

返回参数

VirtualMFADevice

- 类型：[VirtualMFADevice](#)
- 描述：虚拟多因素认证设备

需要的权限

Action

```
ram:CreateVirtualMFADevice
```

Resource

```
acs:ram:*:${AccountId}:mfa/*
```

错误信息

InvalidParameter.VirtualMFADeviceName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "VirtualMFADeviceName" contains invalid chars.

InvalidParameter.VirtualMFADeviceName.Length

- HTTP Status : 400
- Error Message : The parameter - "VirtualMFADeviceName" beyond the length limit.

EntityAlreadyExists.VirtualMFADevice

- HTTP Status : 409
- Error Message : The virtual mfa device does already EXIST.

LimitExceeded.VirtualMFADevice

- HTTP Status : 409
- Error Message : The count of virtual mfa devices beyond the current limits.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=CreateVirtualMFADevice
&VirtualMFADeviceName=device001
&<公共请求参数>
```

返回示例

- XML格式

```
<CreateVirtualMFADeviceResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <VirtualMFADevice>
    <SerialNumber>acs:ram::1234567890123:mfa/device001</
VirtualMFASerialNumber>
```

```
<Base32StringSeed>DSF98HAD982KJA9SDFNAS9D8FU839B8ADHBGSD7A8
</Base32StringSeed>
<QRCodePNG>YXNkZmFzZDlmew5hc2Q5OGZoODd4bXJmcThhaGU5
aSBmYXNkZiBzYWRmIGFGIDRxd2VjIGEdHEzdCAg</QRCodePNG>
</VirtualMFADevice>
</CreateVirtualMFADeviceResponse>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "VirtualMFADevice": {
    "SerialNumber": "acs:ram::1234567890123:mfa/device001",
    "Base32StringSeed": "DSF98HAD982KJA9SDFNAS9D8FU839B
8ADHBGSD7A8",
    "QRCodePNG": "YXNkZmFzZDlmew5hc2Q5OGZoODd4bXJmcThhaGU5
aSBmYXNkZiBzYWRmIGFGIDRxd2VjIGEdHEzdCAg"
  }
}
```

1.4.15 ListVirtualMFADevices

接口描述

列出虚拟多因素认证设备

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：ListVirtualMFADevices

返回参数

VirtualMFADevices

- 类型：[VirtualMFADevice](#) Array
- 描述：虚拟MFA设备列表

需要的权限

Action

```
ram:ListVirtualMFADevices
```

Resource

```
acs:ram:*:${AccountId}:mfa/*
```

操作示例

请求示例

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

返回示例

- XML格式

```
<ListVirtualMFADevicesResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <VirtualMFADevices>
    <VirtualMFADevice>
      <SerialNumber>acs:ram::1234567890123:mfa/device001</
SerialNumber>
    </VirtualMFADevice>
    <VirtualMFADevice>
      <SerialNumber>acs:ram::1234567890123:mfa/device002</
MFASerialNumber>
      <ActivateDate>2015-02-18T17:22:08Z<Activate>
      <User>
        <UserId>1227489245380721</UserId>
        <UserName>zhangqiang</UserName>
        <DisplayName>张强</DisplayName>
      </User>
    </VirtualMFADevice>
  </VirtualMFADevices>
</ListVirtualMFADevicesResponse>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "VirtualMFADevices": {
    "VirtualMFADevice": [
      {
        "SerialNumber": "acs:ram::1234567890123:mfa/
device001"
      },
      {
        "SerialNumber": "acs:ram::1234567890123:mfa/
device002",
        "ActivateDate": "2015-02-18T17:22:08Z",
        "User": {
```

```
{
  "UserId": "1227489245380721",
  "UserName": "zhangqiang",
  "DisplayName": "张强"
}
```

1.4.16 DeleteVirtualMFADevice

接口描述

删除虚拟多因素认证设备

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：DeleteVirtualMFADevice

SerialNumber

- 类型：String
- 必须：是
- 描述：指定MFA设备的序列号

返回参数

只有公共返回参数，详见[公共参数](#)

需要的权限

Action

```
ram:DeleteVirtualMFADevice
```

Resource

```
${SerialNumber}
```

错误信息

EntityNotExist.VirtualMFADevice

- HTTP Status：404

- Error Message : The virtual mfa device does not exist.

DeleteConflict.VirtualMFADevice.User

- HTTP Status : 409
- Error Message : The virtual mfa device CAN NOT been bound to any user while deleting it.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=DeleteVirtualMFADevice
&SerialNumber=acs:ram::1234567890123:mfa/device002
&<公共请求参数>
```

返回示例

- XML格式

```
<DeleteVirtualMFADeviceResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
</DeleteVirtualMFADeviceResponse>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368"
}
```

1.4.17 BindMFADevice

接口描述

绑定多因素认证设备

请求参数

Action

- 类型 : String
- 必须 : 是
- 描述 : 操作接口名，系统规定参数，取值：BindMFADevice

SerialNumber

- 类型 : String
- 必须 : 是
- 描述 : 指定MFA设备的序列号

UserName

- 类型：String
- 必须：是
- 描述：指定用户名

AuthenticationCode1

- 类型：String
- 必须：是
- 描述：验证第一组动态密码

AuthenticationCode2

- 类型：String
- 必须：是
- 描述：验证第二组动态密码

返回参数

只有公共返回参数，详见[公共参数](#)。

需要的权限**Action**

```
ram:BindMFADevice
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息**InvalidParameter.UserName.InvalidChars**

- HTTP Status：400
- Error Message：The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status：400
- Error Message：The parameter - "UserName" beyond the length limit.

InvalidParameter.AuthenticationCode.Format

- HTTP Status : 400
- Error Message : The parameter - "AuthenticationCode" MUST be 6 numbers.

CheckAuthenticationCodeFail

- HTTP Status : 403
- Error Message : The authentication code is incorrect.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

EntityNotExist.VirtualMFADevice

- HTTP Status : 404
- Error Message : The virtual mfa device does not exist.

EntityAlreadyExists.User.MFADevice

- HTTP Status : 409
- Error Message : The user has already bound a MFA device.

EntityAlreadyExists.VirtualMFADevice.User

- HTTP Status : 409
- Error Message : The virtual mfa device does already been bound to a user.

操作示例**请求示例**

```
https://ram.aliyuncs.com/?Action=BindMFADevice
&SerialNumber=acs:ram::1234567890123:mfa/device002
&UserName=zhangqiang
&<公共请求参数>
```

返回示例

- XML格式

```
<BindMFADeviceResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
</BindMFADeviceResponse>
```

- JSON格式

```
{
```

```
} "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368"
```

1.4.18 UnbindMFADevice

接口描述

解绑多因素认证设备

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：UnbindMFADevice

UserName

- 类型：String
- 必须：是
- 描述：指定用户名

返回参数

MFADevice

- 类型：[MFADevice](#)
- 描述：MFA设备信息

需要的权限

Action

```
ram:UnbindMFADevice
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status：400
- Error Message：The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

EntityNotExist.User.MFADevice

- HTTP Status : 404
- Error Message : The user has not bound any MFA device.

操作示例**请求示例**

```
https://ram.aliyuncs.com/?Action=UnbindMFADevice
&UserName=zhangqiang
&<公共请求参数>
```

返回示例

- XML格式

```
<UnbindMFADeviceResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <MFADevice>
    <SerialNumber>acs:ram::1234567890123:mfa/device002</
SerialNumber>
  </MFADevice>
</UnbindMFADeviceResponse>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "MFADevice": {
    "SerialNumber": "acs:ram::1234567890123:mfa/device002"
  }
}
```

1.4.19 GetUserMFAInfo

接口描述

获取指定子用户的多因素认证设备信息

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：GetUserMFAInfo

UserName

- 类型：String
- 必须：是
- 描述：指定用户名

返回参数

MFADevice

- 类型：[MFADevice](#)
- 描述：MFA设备信息

需要的权限

Action

```
ram:GetUserMFAInfo
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

EntityNotExist.User

- HTTP Status：404
- Error Message：The user does not exist.

EntityNotExist.User.MFADevice

- HTTP Status：404
- Error Message：The user has not bound any MFA device.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=GetUserMFAInfo
```

```
&UserName=zhangqiang  
&<公共请求参数>
```

返回示例

- XML格式

```
<GetUserMFAInfoResponse>  
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>  
  <MFADevice>  
    <SerialNumber>Virtual-MFA-device002</SerialNumber>  
  </MFADevice>  
</GetUserMFAInfoResponse>
```

- JSON格式

```
{  
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",  
  "MFADevice": {  
    "SerialNumber": "Virtual-MFA-device002"  
  }  
}
```

1.4.20 ChangePassword

接口描述

修改子用户密码

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：ChangePassword

OldPassword

- 类型：String
- 必须：是
- 描述：旧密码

NewPassword

- 类型：String
- 必须：是

- 描述：指定密码，密码必须符合密码强度要求。关于密码强度设置接口，请参考[SetPasswordPolicy](#)。

返回参数

只有公共返回参数，详见[公共参数](#)

需要的权限

Action

```
ram:ChangePassword
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

NotSupport.Account

- HTTP Status : 400
- Error Message : This method can be only invoked by sub user.

InvalidParameter.OldPassword.Incorrect

- HTTP Status : 400
- Error Message : The parameter - “OldPassword” is incorrect.

InvalidParameter.NewPassword.TooWeak

- HTTP Status : 400
- Error Message : The parameter - “NewPassword” is not compliant with the password policy.

InvalidParameter.NewPassword.ReusePrevention

- HTTP Status : 400
- Error Message : The parameter - “NewPassword” is not compliant with the reuse prevention password policy.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=ChangePassword
&OldPassword=123456
&NewPassword=aw$2ad)d
```

&<公共请求参数>

返回示例

- XML格式

```
<ChangePassword>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
</ChangePassword>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368"
}
```

1.5 组管理接口

1.5.1 CreateGroup

接口描述

创建用户组

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作名称，取值：

CreateGroup

GroupName

- 类型：String
- 必须：是
- 描述：组名称，最大长度64个字符
- 格式：

`^[a-zA-Z0-9\-]+$`

Comments

- 类型：String

- 必须：否
- 描述：备注信息, 最大长度128个字符。

返回参数

Group

- 类型：[Group](#)
- 描述：组信息

需要的权限

Action

```
ram:CreateGroup
```

Resource

```
acs:ram:*:${AccountId}:group/*
```

错误信息

InvalidParameter.GroupName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "GroupName" contains invalid chars.

InvalidParameter.GroupName.Length

- HTTP Status : 400
- Error Message : The parameter - "GroupName" beyond the length limit.

InvalidParameter.Comments.Length

- HTTP Status : 400
- Error Message : The parameter - "Comments" beyond the length limit.

EntityAlreadyExists.Group

- HTTP Status : 409
- Error Message : The group does already EXIST.

LimitExceeded.Group

- HTTP Status : 409
- Error Message : The count of groups beyond the current limits.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=CreateGroup
&GroupName=Dev-Team
&Comments=开发团队
&<公共请求参数>
```

返回示例

- XML格式

```
<CreateGroupResponse>
  <RequestId>D3F0679E-9757-95DB-AF2D-04D5188C69C5</RequestId>
  <Group>
    <GroupName>Dev-Team</GroupName>
    <Comments>开发团队</Comments>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
  </Group>
</CreateGroupResponse>
```

- JSON格式

```
{
  "RequestId": "D3F0679E-9757-95DB-AF2D-04D5188C69C5",
  "Group": {
    "GroupName": "Dev-Team",
    "Comments": "开发团队",
    "CreateDate": "2015-01-23T12:33:18Z"
  }
}
```

1.5.2 GetGroup

接口描述

获取用户组信息

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作名称，取值：

```
GetGroup
```

GroupName

- 类型：String
- 必须：是
- 描述：组名。例:dev
- 格式：

```
^[a-zA-Z0-9\-]+$
```

返回参数

Group

- 类型：[Group](#)
- 描述：组信息

需要的权限

Action

```
ram:GetGroup
```

Resource

```
acs:ram:*:${AccountId}:group/${GroupName}
```

错误信息

InvalidParameter.GroupName.InvalidChars

- HTTP Status：400
- Error Message：The parameter - "GroupName" contains invalid chars.

InvalidParameter.GroupName.Length

- HTTP Status：400
- Error Message：The parameter - "GroupName" beyond the length limit.

EntityNotExist.Group

- HTTP Status：404
- Error Message：The group does not exist.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=GetGroup  
&GroupName=Dev-Team
```

&<公共请求参数>

返回示例

- XML格式

```
<GetGroupResponse>
  <RequestId>D4065824-E422-3ED6-68B1-1AF7D5C7804C</RequestId>
  <Group>
    <GroupName>Dev-Team</GroupName>
    <Comments>开发团队</Comments>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
    <UpdateDate>2015-02-11T03:15:21Z</UpdateDate>
  </Group>
</GetGroupResponse>
```

- JSON格式

```
{
  "RequestId": "D4065824-E422-3ED6-68B1-1AF7D5C7804C",
  "Group": {
    "GroupName": "Dev-Team",
    "Comments": "开发团队",
    "CreateDate": "2015-01-23T12:33:18Z",
    "UpdateDate": "2015-02-11T03:15:21Z"
  }
}
```

1.5.3 UpdateGroup

接口描述

更新用户组信息

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作名称，取值：

UpdateGroup

GroupName

- 类型：String
- 必须：是
- 描述：指定组名称

- 格式：

```
^[a-zA-Z0-9\ - ]+$
```

NewGroupName

- 类型：String
- 必须：否
- 描述：新的组名
- 格式：

```
^[a-zA-Z0-9\ - ]+$
```

NewComments

- 类型：String
- 必须：否
- 描述：新的备注信息, 最大长度128个字符。

返回参数

Group

- 类型：[Group](#)
- 描述：组信息

需要的权限

Action

```
ram:UpdateGroup
```

Resource

```
acs:ram:*:${AccountId}:group/${GroupName}
```

错误信息

InvalidParameter.GroupName.InvalidChars

- HTTP Status：400
- Error Message：The parameter - "GroupName" contains invalid chars.

InvalidParameter.GroupName.Length

- HTTP Status：400

- Error Message : The parameter - "GroupName" beyond the length limit.

InvalidParameter.NewGroupName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "NewGroupName" contains invalid chars.

InvalidParameter.NewGroupName.Length

- HTTP Status : 400
- Error Message : The parameter - "NewGroupName" beyond the length limit.

InvalidParameter.NewComments.Length

- HTTP Status : 400
- Error Message : The parameter - "NewComments" beyond the length limit.

EntityNotExist.Group

- HTTP Status : 404
- Error Message : The group does not exist.

EntityAlreadyExists.Group

- HTTP Status : 409
- Error Message : The group does already EXIST.

操作示例**请求示例**

```
https://ram.aliyuncs.com/?Action=UpdateGroup
&GroupName=Dev-Team
&NewGroupName=NewDev-Team
&<公共请求参数>
```

返回示例

- XML格式

```
<UpdateGroupResponse>
  <RequestId>EC6647CC-0A36-EC7A-BA72-CC81BF3DE182</RequestId>
  <Group>
    <GroupName>NewDev-Team</GroupName>
    <Comments>开发团队</Comments>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
    <UpdateDate>2015-01-23T12:33:18Z</UpdateDate>
  </Group>
```

```
</UpdateGroupResponse>
```

- JSON格式

```
{
  "RequestId": "EC6647CC-0A36-EC7A-BA72-CC81BF3DE182",
  "Group": {
    "GroupName": "NewDev-Team",
    "Comments": "开发团队",
    "CreateDate" : "2015-01-23T12:33:18Z",
    "UpdateDate" : "2015-01-23T12:33:18Z"
  }
}
```

1.5.4 ListGroups

接口描述

列出所有的用户组

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作名称，取值：

```
ListGroups
```

Marker

- 名称：Marker
- 类型：String
- 必须：否
- 描述：当请求的返回结果被截断时，可以使用Marker获取从当前截断位置之后的内容

MaxItems

- 名称：MaxItems
- 类型：Integer
- 必须：否
- 取值范围：[1 - 1000]
- 默认值：100

- 描述：指定返回结果的条数，当返回结果达到MaxItems限制被截断时，返回参数IsTruncated将等于true。

返回参数

Groups

- 类型：Group Array
- 描述：组信息集合

IsTruncated

- 类型：Boolean
- 描述：请求返回结果是否被截断

Marker

- 类型：String
- 描述：当IsTruncated为true时才有此字段，当返回true时，需要继续调用此接口，并且使用Marker获取截断后的内容

需要的权限

Action

```
ram:ListGroup
```

Resource

```
acs:ram:*:${AccountId}:group/*
```

操作示例

请求示例

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

返回示例

- XML格式

```
<ListGroupResponse>
  <RequestId>065527AA-2F2E-AD7C-7484-F2626CFE4934</RequestId>
  <IsTruncated>true</IsTruncated>
  <Marker>EXAMPLE</Marker>
  <Groups>
    <Group>
      <GroupName>Dev-Team</GroupName>
```

```
<Comments>开发团队</Comments>
<CreateDate>2015-01-23T12:33:18Z</CreateDate>
<UpdateDate>2015-01-23T12:33:18Z</UpdateDate>
</Group>
<Group>
  <GroupName>QA-Team</GroupName>
  <Comments>测试团队</Comments>
  <CreateDate>2015-02-18T17:22:08Z</CreateDate>
  <UpdateDate>2015-02-18T17:22:08Z</UpdateDate>
</Group>
</Groups>
</ListGroupResponse>
```

- JSON格式

```
{
  "RequestId": "065527AA-2F2E-AD7C-7484-F2626CFE4934",
  "IsTruncated": true,
  "Marker": "EXAMPLE",
  "Groups": {
    "Group": [
      {
        "GroupName": "Dev-Team",
        "Comments": "开发团队",
        "CreateDate" : "2015-01-23T12:33:18Z",
        "UpdateDate" : "2015-01-23T12:33:18Z"
      },
      {
        "GroupName": "QA-Team",
        "Comments": "测试团队",
        "CreateDate" : "2015-02-18T17:22:08Z",
        "UpdateDate" : "2015-02-18T17:22:08Z"
      }
    ]
  }
}
```

1.5.5 DeleteGroup

接口描述

删除指定的组

- 删除组前，需要保证组不拥有任何授权策略(Policy)
- 删除组前，需要保证组不拥有任何用户(User)

请求参数

Action

- 类型：String
- 必须：是

- 描述：操作名称，取值：

```
DeleteGroup
```

GroupName

- 类型：String
- 必须：是
- 描述：指定组名。例:dev
- 格式：

```
^[a-zA-Z0-9\-]+$
```

返回参数

只有公共返回参数，详见[公共参数](#)

需要的权限

Action

```
ram>DeleteGroup
```

Resource

```
acs:ram:*:${AccountId}:group/${GroupName}
```

错误信息

InvalidParameter.GroupName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "GroupName" contains invalid chars.

InvalidParameter.GroupName.Length

- HTTP Status : 400
- Error Message : The parameter - "GroupName" beyond the length limit.

EntityNotExist.Group

- HTTP Status : 404
- Error Message : The group does not exist.

DeleteConflict.Group.User

- HTTP Status : 409
- Error Message : The group CAN NOT has any user member while deleting the group.

DeleteConflict.Group.Policy

- HTTP Status : 409
- Error Message : The entity CAN NOT has any attached policy while deleting the group.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=DeleteGroup
&GroupName=Dev-Team
&<公共请求参数>
```

返回示例

- XML格式

```
<DeleteGroupResponse>
  <RequestId>FCF40AB5-881C-A0F9-334C-B0AD423AA69D</RequestId>
</DeleteGroupResponse>
```

- JSON格式

```
{
  "RequestId": "FCF40AB5-881C-A0F9-334C-B0AD423AA69D"
}
```

1.5.6 AddUserToGroup

接口描述

将子用户添加到指定的用户组

请求参数

Action

- 类型 : String
- 必须 : 是
- 描述 : 操作名称，取值 :

```
AddUserToGroup
```

UserName

- 类型 : String

- 必须：是
- 描述：用户名称

GroupName

- 类型：String
- 必须：是
- 描述：组名称

返回参数

只有公共返回参数，详见[公共参数](#)

需要的权限**Action**

```
ram:AddUserToGroup
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

```
acs:ram:*:${AccountId}:group/${GroupName}
```

错误信息**InvalidParameter.UserName.InvalidChars**

- HTTP Status : 400
- Error Message : The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

InvalidParameter.GroupName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "GroupName" contains invalid chars.

InvalidParameter.GroupName.Length

- HTTP Status : 400
- Error Message : The parameter - "GroupName" beyond the length limit.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

EntityNotExist.Group

- HTTP Status : 404
- Error Message : The group does not exist.

EntityAlreadyExists.User.Group

- HTTP Status : 409
- Error Message : The user has already joined the group.

LimitExceeded.User.Group

- HTTP Status : 409
- Error Message : The count of groups the target user joined beyond the current limits.

操作示例**请求示例**

```
https://ram.aliyuncs.com/?Action=AddUserToGroup
&UserName=zhangqiang
&GroupName=Dev-Team
&<公共请求参数>
```

返回示例

- XML格式

```
<AddUserToGroupResponse>
  <RequestId>1B968853-B423-63A6-FE1F-45E81BC2AD61</RequestId>
</AddUserToGroupResponse>
```

- JSON格式

```
{
  "RequestId": "1B968853-B423-63A6-FE1F-45E81BC2AD61"
}
```

1.5.7 RemoveUserFromGroup

接口描述

将子用户从用户组中移除

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作名称，取值：

```
RemoveUserFromGroup
```

UserName

- 类型：String
- 必须：是
- 描述：子用户名称

GroupName

- 类型：String
- 必须：是
- 描述：组名称

返回参数

只有公共返回参数，详见[公共参数](#)

需要的权限

Action

```
ram:RemoveUserFromGroup
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
acs:ram:*:${AccountId}:group/${GroupName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status：400
- Error Message：The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

InvalidParameter.GroupName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "GroupName" contains invalid chars.

InvalidParameter.GroupName.Length

- HTTP Status : 400
- Error Message : The parameter - "GroupName" beyond the length limit.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

EntityNotExist.Group

- HTTP Status : 404
- Error Message : The group does not exist.

EntityNotExist.User.Group

- HTTP Status : 404
- Error Message : The user has not joined the group.

操作示例**请求示例**

```
https://ram.aliyuncs.com/?Action=RemoveUserFromGroup
&UserName=zhangqiang
&GroupName=Dev-Team
&<公共请求参数>
```

返回示例

- XML格式

```
<RemoveUserFromGroupResponse>
  <RequestId>A07EF215-B9B3-8CB2-2899-3F9575C6E320</RequestId>
</RemoveUserFromGroupResponse>
```

- JSON格式

```
{
```

```
} "RequestId": "A07EF215-B9B3-8CB2-2899-3F9575C6E320"
```

1.5.8 ListGroupsForUser

接口描述

列出指定子用户所加入的组信息

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作名称，取值：

```
ListGroupsForUser
```

UserName

- 类型：String
- 必须：是
- 描述：用户名称

返回参数

Groups

- 类型：[Group](#) Array
- 描述：组信息集合

需要的权限

Action

```
ram:ListGroupsForUser
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
```

错误信息

InvalidParameter.UserName.InvalidChars

- HTTP Status：400

- Error Message : The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status : 400
- Error Message : The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status : 404
- Error Message : The user does not exist.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=ListGroupsForUser
&UserName=zhangqiang
&<公共请求参数>
```

返回示例

- XML格式

```
<ListGroupsForUserResponse>
  <RequestId>0945BF6F-54A4-C384-C89C-C500DCEE367C</RequestId>
  <Groups>
    <Group>
      <GroupName>Dev-Team</GroupName>
      <Comments>开发团队</Comments>
      <JoinDate>2015-01-23T12:33:18Z</JoinDate>
    </Group>
    <Group>
      <GroupName>QA-Team</GroupName>
      <Comments>测试团队</Comments>
      <JoinDate>2015-02-18T17:22:08Z</JoinDate>
    </Group>
  </Groups>
</ListGroupsForUserResponse>
```

- JSON格式

```
{
  "RequestId": "0945BF6F-54A4-C384-C89C-C500DCEE367C",
  "Groups": {
    "Group": [
      {
        "GroupName": "Dev-Team",
        "Comments": "开发团队",
        "JoinDate": "2015-01-23T12:33:18Z"
      },
      {
        "GroupName": "QA-Team",
        "Comments": "测试团队",

```

```
    "JoinDate" : "2015-02-18T17:22:08Z"
  }
]
}
```

1.5.9 ListUsersForGroup

接口描述

列出指定用户组所包含的子用户。

请求参数

Action

- 类型：String
- 是否必需：是
- 描述：操作名称，取值：

ListUsersForGroup

GroupName

- 类型：String
- 是否必需：是
- 描述：组名。例:Dev-Team

Marker

- 名称：Marker
- 类型：String
- 是否必需：否
- 描述：当请求的返回结果被截断时，可以使用Marker获取从当前截断位置之后的内容

MaxItems

- 名称：MaxItems
- 类型：Integer
- 是否必需：否
- 取值范围：[1 - 1000]
- 默认值：100

- 描述：指定返回结果的条数。当返回结果达到MaxItems限制被截断时，返回参数IsTruncated将等于true。

返回参数

Users

- 类型：User Array
- 描述：用户信息集合

IsTruncated

- 类型：Boolean
- 描述：请求返回结果是否被截断

Marker

- 类型：String
- 描述：当IsTruncated为true时才有此字段。当返回true时，需要继续调用此接口，并且使用Marker获取截断后的内容。

需要的权限

Action

```
ram:ListUsersForGroup
```

Resource

```
acs:ram:*:${AccountId}:group/${GroupName}
```

错误信息

EntityNotExist.Group

- HTTP Status : 404
- Error Message : The group does not exist.

InvalidParameter.GroupName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - “GroupName” contains invalid chars.

InvalidParameter.GroupName.Length

- HTTP Status : 400

- Error Message : The parameter - “GroupName” beyond the length limit.

EntityNotExist.Group

- HTTP Status : 404
- Error Message : The group does not exist.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=ListUsersForGroup
&GroupName=dev
&<公共请求参数>
```

返回示例

- XML格式

```
<ListUsersForGroupResponse>
  <RequestId>5756784B-79C4-E82E-24C2-FC3E171E5AB3</RequestId>
  <Users>
    <User>
      <UserId>1227489245380721</UserId>
      <UserName>zhangqiang</UserName>
      <DisplayName>张强</DisplayName>
      <JoinDate>2015-01-23T12:33:18Z</JoinDate>
    </User>
    <User>
      <UserId>1406498224724456</UserId>
      <UserName>lili</UserName>
      <DisplayName>李丽</DisplayName>
      <JoinDate>2015-02-18T17:22:08Z</JoinDate>
    </User>
  </Users>
</ListUsersForGroupResponse>
```

- JSON格式

```
{
  "RequestId" : "4B450CA1-36E8-4AA2-8461-86B42BF4CC4E",
  "Users" : {
    "User" : [
      {
        "UserId" : "1227489245380721",
        "UserName" : "zhangqiang",
        "DisplayName" : "张强",
        "JoinDate" : "2015-01-23T12:33:18Z"
      },
      {
        "UserId" : "1406498224724456",
        "UserName" : "lili",
        "DisplayName" : "李丽",
        "JoinDate" : "2015-02-18T17:22:08Z"
      }
    ]
  }
}
```

```
}  
}
```

1.6 角色管理接口

1.6.1 CreateRole

接口描述

创建角色

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：CreateRole

RoleName

- 类型：String
- 必须：是
- 描述：指定角色名，最多包含64个字符
- 格式：

```
^[a-zA-Z0-9\.\@\-]+\s$
```

Description

- 类型：String
- 必须：否
- 描述：角色描述，最大长度1024个字符

AssumeRolePolicyDocument

- 类型：String
- 必须：是
- 描述：一个策略文本，指定受信任的允许扮演该角色的一个或多个主体(Principal)，这个主体可以是RAM用户或阿里云服务。
- 样例说明：

- 如下策略表示允许扮演该角色的受信主体为云账号(AccountID=123456789012345678)下被授权的RAM用户:

```
{
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Effect": "Allow",
      "Principal": {
        "RAM": [
          "acs:ram::123456789012345678:root"
        ]
      }
    }
  ],
  "Version": "1"
}
```

- 如下策略表示允许扮演该角色的受信主体为当前租户下的ECS实例 (用户创建ECS实例时可以指定使用该RAM角色, 实例启动后将能获得该RAM角色的STS-Token):

```
{
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "ecs.aliyuncs.com"
        ]
      }
    }
  ],
  "Version": "1"
}
```

返回参数

Role

- 类型: [Role](#)
- 描述: 角色信息

需要的权限

Action

```
ram:CreateRole
```

Resource

```
acs:ram:*:${AccountId}:role/*
```

错误信息

InvalidParameter.RoleName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - “RoleName” contains invalid chars.

InvalidParameter.RoleName.Length

- HTTP Status : 400
- Error Message : The parameter - “RoleName” beyond the length limit.

InvalidParameter.AssumeRolePolicyDocument.Length

- HTTP Status : 400
- Error Message : The parameter - “AssumeRolePolicyDocument” beyond the length limit.

EntityAlreadyExists.Role

- HTTP Status : 409
- Error Message : The role does already EXIST.

MalformedPolicyDocument

- HTTP Status : 400
- Error Message : {The error details}

LimitExceeded.Role

- HTTP Status : 409
- Error Message : The count of roles beyond the current limits.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=CreateRole  
&RoleName=ECSAdmin
```

```
&AssumeRolePolicyDocument={ "Statement": [ { "Action": "sts:AssumeRole", "Effect": "Allow", "Principal": { "RAM": "acs:ram::123456789012345678:root" } } ], "Version": "1" }
&Description=ECS管理角色
&<公共请求参数>
```

返回示例

- XML格式

```
<CreateRoleResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <Role>
    <RoleId>901234567890123</RoleId>
    <RoleName>ECSAdmin</RoleName>
    <Arn>acs:ram::1234567890123456:role/ECSAdmin</Arn>
    <Description>ECS管理角色</Description>
    <AssumeRolePolicyDocument>{ "Statement": [ { "Action": "
sts:AssumeRole", "Effect": "Allow", "Principal": { "RAM": "acs:
ram::123456789012345678:root" } } ], "Version": "1" }</AssumeRole
PolicyDocument>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
  </Role>
</CreateRoleResponse>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "Role": {
    "RoleId": "901234567890123",
    "RoleName": "ECSAdmin",
    "Arn": "acs:ram::1234567890123456:role/ECSAdmin",
    "Description": "ECS管理角色",
    "AssumeRolePolicyDocument": "{ \"Statement\": [ { \"Action
\": \"sts:AssumeRole\", \"Effect\": \"Allow\", \"Principal\": { \"
RAM\": \"acs:ram::123456789012345678:root\" } } ], \"Version\": \"1
\" }",
    "CreateDate": "2015-01-23T12:33:18Z"
  }
}
```

1.6.2 GetRole

接口描述

获取角色信息

请求参数

Action

- 类型：String
- 必须：是

- 描述：操作接口名，系统规定参数，取值：GetRole

RoleName

- 类型：String
- 必须：是
- 描述：指定角色名，最多包含64个字符
- 格式：

```
^[a-zA-Z0-9\.\@\-]+\
```

返回参数**Role**

- 类型：[Role](#)
- 描述：角色信息

需要的权限**Action**

```
ram:GetRole
```

Resource

```
acs:ram:*:${AccountId}:role/*
```

错误信息**InvalidParameter.RoleName.InvalidChars**

- HTTP Status : 400
- Error Message : The parameter - "RoleName" contains invalid chars.

InvalidParameter.RoleName.Length

- HTTP Status : 400
- Error Message : The parameter - "RoleName" beyond the length limit.

EntityNotExist.Role

- HTTP Status : 404
- Error Message : The role does not exist.

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=GetRole
&RoleName=ECSAdmin
&<公共请求参数>
```

返回示例

- XML格式

```
<GetRoleResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <Role>
    <RoleId>901234567890123</RoleId>
    <RoleName>ECSAdmin</RoleName>
    <Arn>acs:ram::1234567890123456:role/ECSAdmin</Arn>
    <Description>ECS管理角色</Description>
    <AssumeRolePolicyDocument>{ "Statement": [ { "Action": "
sts:AssumeRole", "Effect": "Allow", "Principal": { "RAM": "acs:
ram::123456789012345678:root" } } ], "Version": "1" }</AssumeRole
PolicyDocument>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
    <UpdateDate>2015-01-23T12:33:18Z</UpdateDate>
  </Role>
</GetRoleResponse>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "Role": {
    "RoleId": "901234567890123",
    "RoleName": "ECSAdmin",
    "Arn": "acs:ram::1234567890123456:role/ECSAdmin",
    "Description": "ECS管理角色",
    "AssumeRolePolicyDocument": "{ \"Statement\": [ { \"Action
\": \"sts:AssumeRole\", \"Effect\": \"Allow\", \"Principal\": { \"
RAM\": \"acs:ram::123456789012345678:root\" } } ], \"Version\": \"1
\" }",
    "CreateDate": "2015-01-23T12:33:18Z",
    "UpdateDate": "2015-01-23T12:33:18Z"
  }
}
```

1.6.3 UpdateRole

接口描述

更新角色信息

请求参数

Action

- 类型：String
- 必须：是
- 描述：操作接口名，系统规定参数，取值：UpdateRole

RoleName

- 类型：String
- 必须：是
- 描述：指定角色名，最多包含64个字符
- 格式：

```
^[a-zA-Z0-9\.\@\-]+\$\
```

NewAssumeRolePolicyDocument

- 类型：String
- 必须：否
- 描述：指定可以扮演此角色的身份。

返回参数**Role**

- 类型：[Role](#)
- 描述：角色信息

需要的权限**Action**

```
ram:UpdateRole
```

Resource

```
acs:ram:*:${AccountId}:role/*
```

错误信息**InvalidParameter.RoleName.InvalidChars**

- HTTP Status：400
- Error Message：The parameter - "RoleName" contains invalid chars.

InvalidParameter.RoleName.Length

- HTTP Status : 400
- Error Message : The parameter - "RoleName" beyond the length limit.

InvalidParameter.NewAssumeRolePolicyDocument.Length

- HTTP Status : 400
- Error Message : The parameter - "NewAssumeRolePolicyDocument" beyond the length limit.

EntityNotExist.Role

- HTTP Status : 404
- Error Message : The role does not exist.

MalformedPolicyDocument

- HTTP Status : 400
- Error Message : {The error details}

操作示例

请求示例

```
https://ram.aliyuncs.com/?Action=UpdateRole
&RoleName=ECSAdmin
&NewAssumeRolePolicyDocument=xxxx //TODO
&<公共请求参数>
```

返回示例

- XML格式

```
<UpdateRoleResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <Role>
    <RoleId>901234567890123</RoleId>
    <RoleName>ECSAdmin</RoleName>
    <Arn>acs:ram::1234567890123456:role/ECSAdmin</Arn>
    <Description>ECS管理角色</Description>
    <AssumeRolePolicyDocument>{ "Statement": [ { "Action": "
sts:AssumeRole", "Effect": "Allow", "Principal": { "RAM": "acs:
ram::123456789012345678:root" } } ], "Version": "1" }</AssumeRole
PolicyDocument>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
    <UpdateDate>2015-01-23T12:33:18Z</UpdateDate>
  </Role>
</UpdateRoleResponse>
```

- JSON格式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
```

```
"Role": {
  "RoleId": "901234567890123",
  "RoleName": "ECSAdmin",
  "Arn": "acs:ram::1234567890123456:role/ECSAdmin",
  "Description": "ECS管理角色",
  "AssumeRolePolicyDocument": "{ \"Statement\": [ { \"Action\": \"sts:AssumeRole\", \"Effect\": \"Allow\", \"Principal\": { \"RAM\": \"acs:ram::123456789012345678:root\" } } ], \"Version\": \"1\" }\",
  "CreateDate": "2015-01-23T12:33:18Z",
  "UpdateDate": "2015-01-23T12:33:18Z"
}
```

1.6.4 ListRoles

介面描述

列出角色

請求參數

Action

- 類型：String
- 必須：是
- 描述：系統規定參數，取值：ListRoles

返回參數

Roles

- 類型：[Role](#) Array
- 描述：角色名稱列表

需要的許可權

Action

```
ram:ListRoles
```

Resource

```
acs:ram:*:${AccountId}:role/*
```

操作樣本

請求樣本

```
https://ram.aliyuncs.com/?Action=ListRoles
```

<公共请求参数>

返回样本

- XML格式

```
<ListRolesResponse>
  <RequestId>7B8A4E7D-6CFF-471D-84DF-195A7A241ECB</RequestId>
  <Roles>
    <Role>
      <RoleId>901234567890123</RoleId>
      <RoleName>ECSAdmin</RoleName>
      <Arn>acs:ram::1234567890123456:role/ECSAdmin</Arn>
      <Description>ECS管理角色</Description>
      <CreateDate>2015-01-23T12:33:18Z</CreateDate>
      <UpdateDate>2015-01-23T12:33:18Z</UpdateDate>
    </Role>
    <Role>
      <RoleId>901234567890456</RoleId>
      <RoleName>OSSReadonlyAccess</RoleName>
      <Arn>acs:ram::1234567890123456:role/OSSReadonlyAccess</
      <Description>OSS只读访问角色</Description>
      <CreateDate>2015-01-23T12:33:18Z</CreateDate>
      <UpdateDate>2015-01-23T12:33:18Z</UpdateDate>
    </Role>
  </Roles>
</ListRolesResponse>
```

- JSON格式

```
{
  "RequestId": "7B8A4E7D-6CFF-471D-84DF-195A7A241ECB",
  "Roles": {
    "Role": [
      {
        "RoleId": "901234567890123",
        "RoleName": "ECSAdmin",
        "Arn": "acs:ram::1234567890123456:role/ECSAdmin",
        "Description": "ECS管理角色",
        "CreateDate": "2015-01-23T12:33:18Z",
        "UpdateDate": "2015-01-23T12:33:18Z"
      },
      {
        "RoleId": "901234567890456",
        "RoleName": "OSSReadonlyAccess",
        "Arn": "acs:ram::1234567890123456:role/OSSReadonlyAccess",
        "Description": "OSS只读访问角色",
        "CreateDate": "2015-01-23T12:33:18Z",
        "UpdateDate": "2015-01-23T12:33:18Z"
      }
    ]
  }
}
```

```
}
```

1.6.5 DeleteRole

接口描述

删除指定的角色

请求参数

Action

- 类型：String
- 必须：是
- 描述：系统规定参数，取值：DeleteRole

RoleName

- 类型：String
- 必须：是
- 描述：指定角色名称

返回参数

只有公共返回参数，详见[公共参数](#)

需要的权限

Action

```
ram:DeleteRole
```

Resource

```
acs:ram:*:${AccountId}:role/${RoleName}
```

错误信息

InvalidParameter.RoleName.InvalidChars

- HTTP Status：400
- Error Message：The parameter - "RoleNam" contains invalid chars.

InvalidParameter.RoleName.Length

- HTTP Status：400

- Error Message : The parameter - "RoleName" beyond the length limit.

EntityNotExist.Role

- HTTP Status : 404
- Error Message : The role does not exist.

DeleteConflict.Role.Policy

- HTTP Status : 409
- Error Message : The role CAN NOT has any attached policy while deleting the role.

请求示例

```
https://ram.aliyuncs.com/?Action=DeleteRole
&RoleName=ECSAdmin
&<公共请求参数>
```

返回示例

- XML格式

```
<DeleteRoleResponse>
  <RequestId>898FAB24-7509-43EE-A287-086FE4C44394</RequestId>
</DeleteRoleResponse>
```

- JSON格式

```
{
  "RequestId": "898FAB24-7509-43EE-A287-086FE4C44394"
}
```

1.7 授权策略管理接口

1.7.1 CreatePolicy

Interface description

Creates an authorization policy.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is "CreatePolicy".

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name. It consists of a maximum of 128 characters.
- Format:

```
^[a-zA-Z0-9\ -]+$
```

Description

- Type: String
- Required: No
- Description: Authorization policy description. It consists of a maximum of 1,024 characters.

PolicyDocument

- Type: String
- Required: Yes
- Description: Authorization policy content. It consists of a maximum of 2,048 characters.

Return parameters**Policy**

- Type: [Policy](#)
- Description: Authorization policy content.

Required permissions**Action**

```
ram:CreatePolicy
```

Resource

```
acs:ram:*:${AccountId}:policy/*
```

Error messages**InvalidParameter.PolicyName.InvalidChars**

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

InvalidParameter.PolicyDocument.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

InvalidParameter.Description.Length

- HTTP Status: 400
- Error Message: The parameter - "Description" beyond the length limit.

MalformedPolicyDocument

- HTTP Status: 400
- Error Message: {The error details}

EntityAlreadyExists.Policy

- HTTP Status: 409
- Error Message: The group does already EXIST.

LimitExceeded.Policy

- HTTP Status: 409
- Error Message: The count of policy beyond the current limits.

Request example

```
https://ram.aliyuncs.com/?Action=CreatePolicy
&PolicyName=OSS-Administrator
&PolicyDocument={ "Statement": [{ "Action": ["oss:*"], "Effect": "
Allow", "Resource": ["acs:oss:*:*:*"]}], "Version": "1"}
&Description=OSS administrator permission
&<Public request parameters>
```

Return example

- XML format

```
<CreatePolicyResponse>
  <RequestId>9B34724D-54B0-4A51-B34D-4512372FE1BE</RequestId>
  <Policy>
    <PolicyName>OSS-Administrator</PolicyName>
    <PolicyType>Custom</PolicyType>
    <Description>OSS administrator permission</Description>
```

```
<DefaultVersion>v1</DefaultVersion>
<CreateDate>2015-01-23T12:33:18Z</CreateDate>
</Policy>
</CreatePolicyResponse>
```

- JSON format

```
"RequestId": "9B34724D-54B0-4A51-B34D-4512372FE1BE",
"Policy": {
  "PolicyName": "OSS-Administrator",
  "PolicyType": "Custom",
  "Description": "OSS administrator permission",
  "DefaultVersion": "v1",
  "CreateDate": "2015-01-23T12:33:18Z"
}
```

1.7.2 GetPolicy

Interface description

Obtains the information about a specified authorization policy.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is “GetPolicy”.

PolicyType

- Type: String
- Required: Yes
- Description: Authorization policy type. Value: “System” or “Custom”.

PolicyName

- Name: PolicyName
- Type: String
- Required: Yes
- Description: Authorization policy name.

Return parameters

Policy

- Type: [Policy](#)

- Description: Basic information about an authorization policy.

Required permissions

Action

```
ram:GetPolicy
```

Resource

```
acs:ram:*:${AccountId} or system:policy/${PolicyName}
```

Error messages

InvalidParameter.PolicyType

- HTTP Status: 400
- Error Message: The parameter - "PolicyType" is incorrect.

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

EntityNotExist.Policy

- HTTP Status: 404
- Error Message: The policy does not exist.

Request example

```
Https://ram.aliyuncs.com /? Action = getpolicy  
&PolicyName=OSS-Administrator  
&<Public request parameters>
```

Return example

- XML format

```
<GetPolicyResponse>  
  <RequestId>697852FB-50D7-44D9-9774-530C31EAC572</RequestId>  
  <Policy>  
    <PolicyName>OSS-Administrator</PolicyName>  
    <PolicyType>Custom</PolicyType>  
    <Description>OSS administrator permission</Description>
```

```
<DefaultVersion>v1</DefaultVersion>
<CreateDate>2015-01-23T12:33:18Z</CreateDate>
<UpdateDate>2015-01-23T12:33:18Z</UpdateDate>
<AttachmentCount>0</AttachmentCount>
</Policy>
</GetPolicyResponse>
```

- JSON format

```
"RequestId": "697852FB-50D7-44D9-9774-530C31EAC572",
"Policy": {
  "PolicyName": "OSS-Administrator",
  "PolicyType": "Custom",
  "Description": "OSS administrator permission",
  "DefaultVersion": "v1",
  "CreateDate": "2015-01-23T12:33:18Z",
  "UpdateDate": "2015-01-23T12:33:18Z",
  "AttachmentCount": 0
}
```

1.7.3 DeletePolicy

Interface description

Deletes a specified authorization policy.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is “DeletePolicy”.

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

Return parameters

Only public parameters are returned. For details, see [公共参数](#)

Required permissions

Action

```
ram:DeletePolicy
```

Resource

```
acs:ram:*:${AccountId}:policy/${PolicyName}
```

Error messages

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

EntityNotExist.Policy

- HTTP Status: 404
- Error Message: The policy does not exist.

DeleteConflict.Policy.User

- HTTP Status: 409
- Error Message: The policy CAN NOT been attached to any user while deleting the policy.

DeleteConflict.Policy.Group

- HTTP Status: 409
- Error Message: The policy CAN NOT been attached to any group while deleting the policy.

DeleteConflict.Policy.Version

- HTTP Status: 409
- Error Message: The policy CAN NOT has any version except the default version.

Request example

```
https://ram.aliyuncs.com/?Action=DeletePolicy  
&PolicyName=OSS-Administrator
```

```
&<Public request parameters>
```

Return example

- XML format

```
<DeletePolicyResponse>  
  <RequestId>898FAB24-7509-43EE-A287-086FE4C44394</RequestId>  
</DeletePolicyResponse>
```

- JSON format

```
"RequestId": "898FAB24-7509-43EE-A287-086FE4C44394"
```

1.7.4 ListPolicies

Interface description

Lists authorization policies.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is “ListPolicies”.

PolicyType

- Type: String
- Required: No
- Description: Authorization policy type. Value: “System” or “Custom”. If this parameter is not set , all authorization policies will be listed.

Marker

- Name: Marker
- Type: String
- Required: No
- Description: If a return result is intercepted, you can use “Marker” to obtain the followed content in the return result.

MaxItems

- Name: Maxler
- Type: Integer
- Required: No
- Value range: [1 - 1000]
- Default value: 100
- Description: Quantity of return result items. If a return result is intercepted when it reaches the “MaxItems” restriction, the value of the return parameter `IsTruncated` will be `true`.

Return parameters

IsTruncated

- Type: Boolean
- Description: Whether a return result is intercepted.

Marker

- Type: String
- Description: This parameter is available only when the value of the parameter `IsTruncated` is `true`. If the value is `true`, you need to invoke this interface and use `Marker` to obtain the content following the intercepted part in the return result.

Policies

- Type: [Policy](#) Array
- Description: Authorization policy name list.

Required permissions

Action

```
ram:ListPolicies
```

Resource

```
acs:ram:*:${AccountId}:policy/*
```

Request example

```
https://ram.aliyuncs.com/?Action=ListPolicies
```

&<Public request parameters>

Return example

- XML format

```
<ListPoliciesResponse>
  <RequestId>7B8A4E7D-6CFF-471D-84DF-195A7A241ECB</RequestId>
  <IsTruncated>true</IsTruncated>
  <Marker>EXAMPLE</Marker>
  <Policies>
    <Policy>
      <PolicyName>OSS-Administrator</PolicyName>
      <PolicyType>Custom</PolicyType>
      <Description>OSS administrator permission</Description>
      <DefaultVersion>v1</DefaultVersion>
      <CreateDate>2015-01-23T12:33:18Z</CreateDate>
      <UpdateDate>2015-01-23T12:33:18Z</UpdateDate>
      <AttachmentCount>0</AttachmentCount>
    </Policy>
    <Policy>
      <PolicyName>ReadOnlyAccess</PolicyName>
      <PolicyType>System</PolicyType>
      <Description>ReadOnlyAccess</Description>
      <DefaultVersion>v1</DefaultVersion>
      <CreateDate>2015-02-11T18:39:12Z</CreateDate>
      <UpdateDate>2015-02-19T09:43:16Z</UpdateDate>
      <AttachmentCount>0</AttachmentCount>
    </Policy>
  </Policies>
</ListPoliciesResponse>
```

- JSON format

```
"RequestId": "7B8A4E7D-6CFF-471D-84DF-195A7A241ECB",
"IsTruncated": true,
"Marker": "EXAMPLE",
"Policies": {
  "Policy": [
    {
      "PolicyName": "OSS-Administrator",
      "PolicyType": "Custom",
      "Description": "OSS administrator permission",
      "DefaultVersion": "v1",
      "CreateDate": "2015-01-23T12:33:18Z",
      "UpdateDate": "2015-01-23T12:33:18Z",
      "AttachmentCount": 3
    },
    {
      "PolicyName": "ReadOnlyAccess",
      "PolicyType": "System",
      "Description": "Read-only permission",
      "DefaultVersion": "v1",
      "CreateDate": "2015-02-11T18:39:12Z",
      "UpdateDate": "2015-02-19T09:43:16Z",
      "AttachmentCount": 1
    }
  ]
}
```

1.7.5 CreatePolicyVersion

Interface description

Creates an authorization policy version.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is “CreatePolicyVersion”.

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

PolicyDocument

- Type: String
- Required: Yes
- Description: Authorization policy content. It consists of a maximum of 2,048 characters.

SetAsDefault

- Type: String
- Required: No
- Description: Whether to set as the default authorization policy. The default value is “false”.

Return parameters

PolicyVersion

- Type: [PolicyVersion](#)
- Description: Information about the newly created authorization policy version.

Required permissions

Action

```
ram:CreatePolicyVersion
```

Resource

```
acs:ram:*:${AccountId}:policy/${PolicyName}
```

Error messages

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

EntityNotExist.Policy

- HTTP Status: 404
- Error Message: The policy does not exist.

LimitExceeded.Policy.Version

- HTTP Status: 409
- Error Message: The count of policy version beyond the current limits.

Request example

```
https://ram.aliyuncs.com/?Action=CreatePolicyVersion
&PolicyName=OSS-Administrator
&Policydocument={"statement":[{"action":["Oss: *"],"effect":"
allow","resource":["ACS: OSS: *"]}], "version": "1"}
&<Public request parameters>
```

Return example

- XML format

```
<CreatePolicyVersionResponse>
  <RequestId>9B34724D-54B0-4A51-B34D-4512372FE1BE</RequestId>
  <PolicyVersion>
    <VersionId>v3</VersionId>
    <IsDefaultVersion>false</IsDefaultVersion>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
```

```
</Policy>
</CreatePolicyVersionResponse>
```

- JSON format

```
"RequestId": "9B34724D-54B0-4A51-B34D-4512372FE1BE",
"PolicyVersion": {
  "VersionId": "v3",
  "IsDefaultVersion": false,
  "CreateDate": "2015-01-23T12:33:18Z"
}
```

1.7.6 GetPolicyVersion

Interface description

Obtains an authorization policy version.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is “GetPolicyVersion”.

PolicyType

- Type: String
- Required: Yes
- Description: Authorization policy type. Value: “System” or “Custom”.

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

VersionId

- Type: String
- Required: Yes
- Description: Target version ID.

Return parameters

PolicyVersion

- Name: *PolicyVersion*
- Type: String
- Description: Authorization policy version information.

Required permissions

Action

```
ram:GetPolicyVersion
```

Resource

```
acs:ram:*:${AccountId} or system:group/${PolicyName}
```

Error messages

InvalidParameter.PolicyType

- HTTP Status: 400
- Error Message: The parameter - "PolicyType" is incorrect.

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

InvalidParameter.VersionId.Format

- HTTP Status: 400
- Error Message: The format of the parameter - "VersionId" is incorrect.

EntityNotExist.Policy

- HTTP Status: 404
- Error Message: The policy does not exist.

EntityNotExist.Policy.Version

- HTTP Status: 404
- Error Message: The policy version does not exist.

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=GetPolicyVersion
&PolicyName=OSS-Administrator
&PolicyType=Custom
&VersionId=v3
&<Public request parameters>
```

Return example

- XML format

```
<GetPolicyVersionResponse>
  <RequestId>9B34724D-54B0-4A51-B34D-4512372FE1BE</RequestId>
  <PolicyVersion>
    <VersionId>v3</VersionId>
    <IsDefaultVersion>false</IsDefaultVersion>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
    <PolicyDocument>

      </PolicyDocument>
    </PolicyVersion>
  </GetPolicyVersionResponse>
```

- JSON format

```
{
  "RequestId": "9B34724D-54B0-4A51-B34D-4512372FE1BE",
  "PolicyVersion": {
    "VersionId": "v3",
    "IsDefaultVersion": false,
    "CreateDate": "2015-01-23T12:33:18Z",
    "PolicyDocument": "{ \"Statement\": [{ \"Action\": [\"oss:*\"], \"Effect\": \"Allow\", \"Resource\": [\"acs:oss:*:*:*\" ]}], \"Version\": \"1\"}"
  }
}
```

1.7.7 DeletePolicyVersion

Interface description

Deletes a specified authorization policy version.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is “DeletePolicyVersion”.

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

VersionId

- Type: String
- Required: Yes
- Description: Target version ID.

Return parameters

Only public parameters are returned. For details, see [公共参数](#)

Required permissions**Action**

```
ram:DeletePolicyVersion
```

Resource

```
acs:ram:*:${AccountId}:policy/${PolicyName}
```

Error messages**InvalidParameter.PolicyName.InvalidChars**

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

InvalidParameter.VersionId.Format

- HTTP Status: 400
- Error Message: The format of the parameter - "VersionId" is incorrect.

EntityNotExist.Policy

- HTTP Status: 404
- Error Message: The policy does not exist.

EntityNotExist.Policy.Version

- HTTP Status: 404
- Error Message: The policy version does not exist.

DeleteConflict.Policy.Version.Default

- HTTP Status: 409
- Error Message: The default policy version CAN NOT been deleted directly.

Operation examples**Request example**

```
https://ram.aliyuncs.com/?Action=DeletePolicyVersion
&PolicyName=OSS-Administrator
&VersionId=v3
&<Public request parameters>
```

Return example

- XML format

```
<DeletePolicyVersionResponse>
  <RequestId>9B34724D-54B0-4A51-B34D-4512372FE1BE</RequestId>
</DeletePolicyVersionResponse>
```

- JSON format

```
"RequestId": "9B34724D-54B0-4A51-B34D-4512372FE1BE"
```

1.7.8 ListPolicyVersions

Interface description

Lists authorization policy versions.

Request parameters**Action**

- Type: String
- Required: Yes
- Description: Required. The parameter value is “ListPolicyVersions”.

PolicyType

- Type: String

- Required: Yes
- Description: Authorization policy type. Value: “System” or “Custom”.

PolicyName

- Type: String
- Required: Yes
- Description: Authorization Policy Name

Return parameters**PolicyVersions**

- Type: [PolicyVersion](#) Array
- Description: Authorization policy name list.

Required permissions**Action**

```
ram:ListPolicyVersions
```

Resource

```
acs:ram:*:${AccountId} or system:policy/${PolicyName}
```

Error messages**InvalidParameter.PolicyType**

- HTTP Status: 400
- Error Message: The parameter - "PolicyType" is incorrect.

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

EntityNotExist.Policy

- HTTP Status: 404

- Error Message: The policy does not exist.

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=ListPolicyVersions
&<Public request parameters>
```

Return example

- XML format

```
<ListPolicyVersionsResponse>
  <RequestId>7B8A4E7D-6CFF-471D-84DF-195A7A241ECB</RequestId>
  <PolicyVersions>
    <PolicyVersion>
      <VersionId>v3</VersionId>
      <IsDefaultVersion>false</IsDefaultVersion>
      <CreateDate>2015-01-23T12:33:18Z</CreateDate>
    </PolicyVersion>
    <PolicyVersion>
      <VersionId>v5</VersionId>
      <IsDefaultVersion>true</IsDefaultVersion>
      <CreateDate>2015-02-26T01:25:52Z</CreateDate>
    </PolicyVersion>
  </PolicyVersions>
</ListPolicyVersionsResponse>
```

- JSON format

```
"RequestId": "7B8A4E7D-6CFF-471D-84DF-195A7A241ECB",
"Policyversions ":{
  "PolicyVersion": [
    {
      "VersionId": "v3",
      "IsDefaultVersion": false,
      "CreateDate": "2015-01-23T12:33:18Z"
    },
    {
      "VersionId": "v5",
      "IsDefaultVersion": true,
      "CreateDate": "2015-02-26T01:25:52Z"
    }
  ]
}
```

1.7.9 SetDefaultPolicyVersion

Interface description

Sets the default authorization policy version.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is "SetDefaultPolicyVersion".

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

VersionId

- Type: String
- Required: Yes
- Description: ID of the new default version.

Return parameters

Only public parameters are returned. For details, see [公共参数](#)

Required permissions

Action

```
ram:SetDefaultPolicyVersion
```

Resource

```
acs:ram:*:${AccountId}:policy/${PolicyName}
```

Error messages

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

InvalidParameter.VersionId.Format

- HTTP Status: 400
- Error Message: The format of the parameter - "VersionId" is incorrect.

EntityNotExist.Policy

- HTTP Status: 404
- Error Message: The policy does not exist.

EntityNotExist.Policy.Version

- HTTP Status: 404
- Error Message: The policy version does not exist.

Operation examples**Request example**

```
https://ram.aliyuncs.com/?Action=SetDefaultPolicyVersion
&PolicyName=OSS-Administrator
&VersionId=v2
&<Public request parameters>
```

Return example

- XML format

```
<SetDefaultPolicyVersionResponse>
  <RequestId>9B34724D-54B0-4A51-B34D-4512372FE1BE</RequestId>
</SetDefaultPolicyVersionResponse>
```

- JSON format

```
"RequestId": "9B34724D-54B0-4A51-B34D-4512372FE1BE"
```

1.7.10 AttachPolicyToUser

Interface description

Attaches an authorization policy to a specified user.

Request parameters**Action**

- Type: String
- Required: Yes

- Description: Required. The parameter value is “AttachPolicyToUser.”

PolicyType

- Type: String
- Required: Yes
- Description: Authorization policy type. Value: “System” or “Custom.”

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

UserName

- Type: String
- Required: Yes
- Description: User name. Example: zhangqiang

Return parameters

Only public parameters are returned. For details, see [公共参数](#)

Required permissions**Action**

```
ram:AttachPolicyToUser
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}
acs:ram:*:${AccountId} or system:policy/${PolicyName}
```

Error messages**InvalidParameter.UserName.InvalidChars**

- HTTP Status: 400
- Error Message: The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status: 400

- Error Message: The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status: 404
- Error Message: The user does not exist.

InvalidParameter.PolicyType

- HTTP Status: 400
- Error Message: The parameter - "PolicyType" is incorrect.

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

EntityNotExist.Policy

- HTTP Status: 404
- Error Message: The policy does not exist.

EntityAlreadyExists.User.Policy

- HTTP Status: 409
- Error Message: The user has already been attached this policy.

LimitExceeded.User.Policy

- HTTP Status: 409
- Error Message: The policy count of the user attached policies beyond the current limits.

Operation examples**Request example**

```
https://ram.aliyuncs.com/?Action=AttachPolicyToUser
&PolicyType=Custom
&PolicyName=OSS-Administrator
&UserName=zhangqiang
```

```
&<Public request parameters>
```

Return example

- XML format

```
<AttachPolicyToUserResponse>  
  <RequestId>697852FB-50D7-44D9-9774-530C31EAC572</RequestId>  
</AttachPolicyToUserResponse>
```

- JSON format

```
"RequestId": "697852FB-50D7-44D9-9774-530C31EAC572"
```

1.7.11 DetachPolicyFromUser

Interface description

Detaches a specified authorization policy from a user.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is “DetachPolicyFromUser”.

PolicyType

- Type: String
- Required: Yes
- Description: Authorization policy type. Value: “System” or “Custom”.

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

UserName

- Type: String
- Required: Yes
- Description: User name. Example: zhangqiang

Return parameters

Only public parameters are returned. For details, see [公共参数](#)

Required permissions

Action

```
ram:DetachPolicyFromUser
```

Resource

```
acs:ram:*:${AccountId}:user/${UserName}  
acs:ram:*:${AccountId} or system:policy/${PolicyName}
```

Error messages

InvalidParameter.PolicyType

- HTTP Status: 400
- Error Message: The parameter - "PolicyType" is incorrect.

InvalidParameter.UserName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status: 400
- Error Message: The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status: 404
- Error Message: The user does not exist.

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

EntityNotExist.Policy

- HTTP Status: 404
- Error Message: The policy does not exist.

EntityNotExist.User.Policy

- HTTP Status: 404
- Error Message: The indicate policy of the user does not exist.

Operation examples**Request example**

```
https://ram.aliyuncs.com/?Action=DetachPolicyFromUser
&PolicyType=Custom
&PolicyName=OSS-Administrator
&UserName=zhangqiang
&<Public request parameters>
```

Return example

- XML format

```
<DetachPolicyFromUserResponse>
  <RequestId>697852FB-50D7-44D9-9774-530C31EAC572</RequestId>
</DetachPolicyFromUserResponse>
```

- JSON format

```
"RequestId": "697852FB-50D7-44D9-9774-530C31EAC572"
```

1.7.12 AttachPolicyToGroup

Interface description

Attaches an authorization policy to a specified user group.

Request parameters**Action**

- Type: String
- Required: Yes
- Description: Required. The parameter value is “AttachPolicyToGroup”.

PolicyType

- Type: String

- Required: Yes
- Description: Authorization policy type. Value: “System” or “Custom”.

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

GroupName

- Type: String
- Required: Yes
- Description: Group name. Example: dev.

Return parameters

Only public parameters are returned. For details, see [公共参数](#)

Required permissions**Action**

```
ram:AttachPolicyToGroup
```

Resource

```
acs:ram:*:${AccountId}:group/${GroupName}  
acs:ram:*:${AccountId} or system:policy/${PolicyName}
```

Error messages**InvalidParameter.GroupName.InvalidChars**

- HTTP Status: 400
- Error Message: The parameter - “GroupName” contains invalid chars.

InvalidParameter.GroupName.Length

- HTTP Status: 400
- Error Message: The parameter - “GroupName” beyond the length limit.

InvalidParameter.PolicyType

- HTTP Status: 400

- Error Message: The parameter - “PolicyType” is incorrect.

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - “PolicyNam” contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - “PolicyName” beyond the length limit.

EntityNotExist.Group

- HTTP Status: 404
- Error Message: The group does not exist.

EntityNotExist.Policy

- HTTP Status: 404
- Error Message: The policy does not exist.

EntityAlreadyExists.Group.Policy

- HTTP Status: 409
- Error Message: The group has already been attached this policy.

LimitExceeded.Group.Policy

- HTTP Status: 409
- Error Message: The policy count of the group beyond the current limits.

Operation examples**Request example**

```
https://ram.aliyuncs.com/?Action=AttachPolicyToGroup
&PolicyType=Custom
&PolicyName=OSS-Administrator
&GroupName=dev
&<Public request parameters>
```

Return example

- XML format

```
<AttachPolicyToGroupResponse>
  <RequestId>697852FB-50D7-44D9-9774-530C31EAC572</RequestId>
```

```
</AttachPolicyToGroupResponse>
```

- JSON format

```
"RequestId": "697852FB-50D7-44D9-9774-530C31EAC572"
```

1.7.13 DetachPolicyFromGroup

Interface description

Detaches a specified authorization policy from a group.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is “DetachPolicyFromGroup.”

PolicyType

- Type: String
- Required: Yes
- Description: Authorization policy type. Value: “System” or “Custom.”

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

GroupName

- Type: String
- Required: Yes
- Description: Group name. Example: dev

Return parameters

Only public parameters are returned. For details, see [公共参数](#)

Required permissions

Action

```
ram:DetachPolicyFromGroup
```

Resource

```
acs:ram:*:${AccountId}:group/${GroupName}  
acs:ram:*:${AccountId} or system:policy/${PolicyName}
```

Error messages

InvalidParameter.PolicyType

- HTTP Status: 400
- Error Message: The parameter - "PolicyType" is incorrect.

InvalidParameter.GroupName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "GroupName" contains invalid chars.

InvalidParameter.GroupName.Length

- HTTP Status: 400
- Error Message: The parameter - "GroupName" beyond the length limit.

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

EntityNotExist.Group

- HTTP Status: 404
- Error Message: The group does not exist.

EntityNotExist.Policy

- HTTP Status: 404

- Error Message: The policy does not exist.

EntityNotExist.Group.Policy

- HTTP Status: 404
- Error Message: The indicate policy attached to the group does not exist.

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=DetachPolicyFromGroup
&PolicyType=Custom
&PolicyName=OSS-Administrator
&GroupName=dev
&<Public request parameters>
```

Return example

- XML format

```
<DetachPolicyFromGroupResponse>
  <RequestId>697852FB-50D7-44D9-9774-530C31EAC572</RequestId>
</DetachPolicyFromGroupResponse>
```

- JSON format

```
"RequestId": "697852FB-50D7-44D9-9774-530C31EAC572"
```

1.7.14 AttachPolicyToRole

Interface description

Attaches an authorization policy to a specified role.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is “AttachPolicyToRole”.

PolicyType

- Type: String
- Required: Yes
- Description: Authorization policy type. Value: “System” or “Custom”.

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

RoleName

- Type: String
- Required: Yes
- Description: Role name. Example: dev.

Return parameters

Only public parameters are returned. For details, see [公共参数](#)

Required permissions**Action**

```
ram:AttachPolicyToRole
```

Resource

```
acs:ram:*:${AccountId}:role/${RoleName}  
acs:ram:*:${AccountId} or system:policy/${PolicyName}
```

Error messages**InvalidParameter.RoleName.InvalidChars**

- HTTP Status: 400
- Error Message: The parameter - "RoleName" contains invalid chars.

InvalidParameter.RoleName.Length

- HTTP Status: 400
- Error Message: The parameter - "RoleName" beyond the length limit.

InvalidParameter.PolicyType

- HTTP Status: 400
- Error Message: The parameter - "PolicyType" is incorrect.

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

EntityNotExist.Role

- HTTP Status: 404
- Error Message: The role does not exist.

EntityNotExist.Policy

- HTTP Status: 404
- Error Message: The policy does not exist.

EntityAlreadyExists.Role.Policy

- HTTP Status: 409
- Error Message: The role has already been attached this policy.

LimitExceeded.Role.Policy

- HTTP Status: 409
- Error Message: The policy count of the role beyond the current limits.

Operation examples**Request example**

```
https://ram.aliyuncs.com/?Action=AttachPolicyToRole
&PolicyType=Custom
&PolicyName=OSS-Administrator
&RoleName=OSSAdminRole
&<Public request parameters>
```

Return example

- XML format

```
<AttachPolicyToRoleResponse>
  <RequestId>697852FB-50D7-44D9-9774-530C31EAC572</RequestId>
```

```
</AttachPolicyToRoleResponse>
```

- JSON format

```
"RequestId": "697852FB-50D7-44D9-9774-530C31EAC572"
```

1.7.15 DetachPolicyFromRole

Interface description

Detaches a specified authorization policy from a role.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is “DetachPolicyFromRole.”

PolicyType

- Type: String
- Required: Yes
- Description: Authorization policy type. Value: “System” or “Custom.”

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

RoleName

- Type: String
- Required: Yes
- Description: Role name. Example: dev.

Return parameters

Only public parameters are returned. For details, see [公共参数](#)

Required permissions

Action

```
ram:DetachPolicyFromRole
```

Resource

```
acs:ram:*:${AccountId}:role/${RoleName}  
acs:ram:*:${AccountId} or system:policy/${PolicyName}
```

Error messages

InvalidParameter.RoleName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "RoleName" contains invalid chars.

InvalidParameter.RoleName.Length

- HTTP Status: 400
- Error Message: The parameter - "RoleName" beyond the length limit.

InvalidParameter.PolicyType

- HTTP Status: 400
- Error Message: The parameter - "PolicyType" is incorrect.

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - "PolicyName" beyond the length limit.

EntityNotExist.Role

- HTTP Status: 404
- Error Message: The role does not exist.

EntityNotExist.Policy

- HTTP Status: 404

- Error Message: The policy does not exist.

EntityNotExist.Role.Policy

- HTTP Status: 404
- Error Message: The indicate policy attached to the role does not exist.

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=DetachPolicyFromRole
&PolicyType=Custom
&PolicyName=OSS-Administrator
&RoleName=OSSAdminRole
&<Public request parameters>
```

Return example

- XML format

```
<DetachPolicyFromRoleResponse>
  <RequestId>697852FB-50D7-44D9-9774-530C31EAC572</RequestId>
</DetachPolicyFromRoleResponse>
```

- JSON format

```
"RequestId": "697852FB-50D7-44D9-9774-530C31EAC572"
```

1.7.16 ListEntitiesForPolicy

Interface description

Lists entities referencing an authorization policy.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is “ListEntitiesForPolicy”.

PolicyType

- Type: String
- Required: Yes
- Description: Authorization policy type. Value: “System” or “Custom”.

PolicyName

- Type: String
- Required: Yes
- Description: Authorization policy name.

Return parameters**Groups**

- Type: [Group](#) Array
- Description: Group information set.

Users

- Type: [User](#) Array
- Description: User information set.

Roles

- Type: [Role](#) Array
- Description: Role information set.

Required permissions**Action**

```
ram:ListEntitiesForPolicy
```

Resource

```
acs:ram:*:${AccountId}:group/${PolicyName}
```

Error messages**InvalidParameter.PolicyType**

- HTTP Status: 400
- Error Message: The parameter - “PolicyType” is incorrect.

InvalidParameter.PolicyName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - “PolicyNam” contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status: 400
- Error Message: The parameter - “PolicyName” beyond the length limit.

EntityNotExist.Policy

- HTTP Status: 404
- Error Message: The policy does not exist.

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=ListEntitiesForPolicy
&PolicyName=OSS-Administrator
&PolicyType=Custom
&<Public request parameters>
```

Return example

- XML format

```
<ListEntitiesForPolicyResponse>
  <RequestId>7B8A4E7D-6CFF-471D-84DF-195A7A241ECB</RequestId>
  <Groups>
    <Group>
      <GroupName>QA-Team</GroupName>
      <Comments>Test team</Comments>
      <AttachDate>2015-01-23T12:33:18Z</AttachDate>
    </Group>
    <Group>
      <GroupName>Dev-Team</GroupName>
      <Comments>Development Team</comments>
      <AttachDate>2015-02-18T17:22:08Z</AttachDate>
    </Group>
  </Groups>
  <Users>
    <User>
      <UserId>1227489245380721</UserId>
      <UserName>zhangqiang</UserName>
      <DisplayName>zhangqiang</DisplayName>
      <AttachDate>2015-01-23T12:33:18Z</AttachDate>
    </User>
    <User>
      <UserId>1406498224724456</UserId>
      <UserName>lili</UserName>
      <DisplayName>lili</DisplayName>
      <AttachDate>2015-02-18T17:22:08Z</AttachDate>
    </User>
  </Users>
  <Roles>
    <Role>
      <RoleId>901234567890123</RoleId>
      <RoleName>ECSAdmin</RoleName>
      <Arn>acs:ram::1234567890123456:role/ECSAdmin</Arn>
      <Description>ECS Administrator role</Description>
      <AttachDate>2015-01-23T12:33:18Z</AttachDate>
```

```

        </Role>
        <Role>
            <RoleId>901234567890456</RoleId>
            <RoleName>OSSReadonlyAccess</RoleName>
            <Description>OSS read-only access role</Description>
            <Arn>acs:ram::1234567890123456:role/OSSReadonlyAccess</
Arn>
            <AttachDate>2015-01-23T12:33:18Z</AttachDate>
        </Role>
    </Roles>
</ListEntitiesForPolicyResponse>

```

- JSON format

```

"RequestId" : "7B8A4E7D-6CFF-471D-84DF-195A7A241ECB",
"Groups" : {
    "Group" : [

        "GroupName": "QA-Team",
        "Comments": "Test team",
        "AttachDate" : "2015-01-23T12:33:18Z"

        "GroupName": "Dev-Team",
        "Comments": "Development team",
        "AttachDate" : "2015-02-18T17:22:08Z"

    ]
}
"Users" : {
    "User" : [

        "UserId" : "1227489245380721",
        "UserName" : "zhangqiang",
        "DisplayName": "zhangqiang",
        "AttachDate" : "2015-01-23T12:33:18Z"

        "UserId" : "1406498224724456",
        "UserName" : "lili",
        "DisplayName": "lili",
        "AttachDate" : "2015-02-18T17:22:08Z"

    ]
}
"Roles" : {
    "Role" : [

        "RoleId" : "1227489245380721",
        "RoleName" : "ECSAdmin",
        "Arn": "acs:ram::1234567890123456:role/ECSAdmin",
        "Description": "ECS Administrator role",
        "AttachDate" : "2015-01-23T12:33:18Z"

        "RoleId" : "1406498224724456",
        "RoleName" : "OSSReadonlyAccess",
        "Arn": "acs:ram::1234567890123456:role/OSSReadonlyAccess",
        "Description": "OSS administrator read-only permission",
    ]
}

```

```
"AttachDate" : "2015-02-18T17:22:08Z"
```

1.7.17 ListPoliciesForUser

Interface description

Lists authorization policies granted to a user.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is "ListPoliciesForUser."

UserName

- Type: String
- Required: Yes
- Description: User name. Example: zhangqiang

Return parameters

Policies

- Type: [Policy](#) Array
- Description: Authorization policy name list.

Error messages

InvalidParameter.UserName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status: 400
- Error Message: The parameter - "UserName" beyond the length limit.

EntityNotExist.User

- HTTP Status: 404
- Error Message: The user does not exist.

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=ListPoliciesForUser
&<Public request parameters>
```

Return example

- XML format

```
<ListPoliciesForUserResponse>
  <RequestId>7B8A4E7D-6CFF-471D-84DF-195A7A241ECB</RequestId>
  <Policies>
    <Policy>
      <PolicyName>OSS-Administrator</PolicyPolicyName>
      <PolicyType>Custom</PolicyType>
      <Description>OSS administrator permission</Description>
      <DefaultVersion>v1</DefaultVersion>
      <AttachDate>2015-01-23T12:33:18Z</AttachDate>
    </Policy>
    <Policy>
      <PolicyName>ReadOnlyAccess</PolicyPolicyName>
      <PolicyType>System</PolicyType>
      <Description>ReadOnlyAccess</Description>
      <DefaultVersion>v1</DefaultVersion>
      <AttachDate>2015-02-18T17:22:08Z</AttachDate>
    </Policy>
  </Policies>
</ListPoliciesForUserResponse>
```

- JSON format

```
"RequestId": "7B8A4E7D-6CFF-471D-84DF-195A7A241ECB",
"Policies": {
  "Policy": [
    {
      "PolicyName": "OSS-Administrator",
      "PolicyType": "Custom",
      "Description": "OSS administrator permission",
      "DefaultVersion": "v1",
      "AttachDate": "2015-01-23T12:33:18Z"
    },
    {
      "PolicyName": "ReadOnlyAccess",
      "PolicyType": "System",
      "Description": "Read-only permission",
      "DefaultVersion": "v1",
      "AttachDate": "2015-02-18T17:22:08Z"
    }
  ]
}
```

1.7.18 ListPoliciesForGroup

Interface description

Lists authorization policies granted to a group.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is "ListPoliciesForGroup."

GroupName

- Type: String
- Required: Yes
- Description: Group name. Example: dev.

Return parameters

Policies

- Type: *Policy* Array
- Description: Authorization policy name list.

Error messages

InvalidParameter.GroupName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "GroupName" contains invalid chars.

InvalidParameter.GroupName.Length

- HTTP Status: 400
- Error Message: The parameter - "GroupName" beyond the length limit.

EntityNotExist.Group

- HTTP Status: 404
- Error Message: The group does not exist.

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=ListPoliciesForGroup
&<Public request parameters>
```

Return example

- XML format

```
<ListPoliciesForGroupResponse>
  <RequestId>7B8A4E7D-6CFF-471D-84DF-195A7A241ECB</RequestId>
  <Policies>
    <Policy>
      <PolicyName>OSS-Administrator</PolicyName>
      <PolicyType>Custom</PolicyType>
      <Description>OSS administrator permission</Description>
      <DefaultVersion>v1</DefaultVersion>
      <AttachDate>2015-01-23T12:33:18Z</AttachDate>
    </Policy>
    <Policy>
      <PolicyName>ReadOnlyAccess</PolicyName>
      <PolicyType>System</PolicyType>
      <Description>ReadOnlyAccess</Description>
      <DefaultVersion>v1</DefaultVersion>
      <AttachDate>2015-02-18T17:22:08Z</AttachDate>
    </Policy>
  </Policies>
</ListPoliciesForGroupResponse>
```

- JSON format

```
"RequestId": "7B8A4E7D-6CFF-471D-84DF-195A7A241ECB",
"Policies": {
  "Policy": [
    {
      "PolicyName": "OSS-Administrator",
      "PolicyType": "Custom",
      "Description": "OSS administrator permission",
      "DefaultVersion": "v1",
      "AttachDate": "2015-01-23T12:33:18Z"
    },
    {
      "PolicyName": "ReadOnlyAccess",
      "PolicyType": "System",
      "Description": "Read-only permission",
      "DefaultVersion": "v1",
      "AttachDate": "2015-02-18T17:22:08Z"
    }
  ]
}
```

1.7.19 ListPoliciesForRole

Interface description

Lists authorization policies granted to a role.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Required. The parameter value is "ListPoliciesForRole".

RoleName

- Type: String
- Required: Yes
- Description: Role name. Example: AdminRole.

Return parameters

Policies

- Type: [Policy](#) Array
- Description: Authorization policy name list.

Error messages

InvalidParameter.RoleName.InvalidChars

- HTTP Status: 400
- Error Message: The parameter - "RoleName" contains invalid chars.

InvalidParameter.RoleName.Length

- HTTP Status: 400
- Error Message: The parameter - "RoleName" beyond the length limit.

EntityNotExist.Role

- HTTP Status: 404
- Error Message: The role does not exist.

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=ListPoliciesForRole
&<Public request parameters>
```

Return example

- XML format

```
<ListPoliciesForRoleResponse>
  <RequestId>7B8A4E7D-6CFF-471D-84DF-195A7A241ECB</RequestId>
  <Policies>
    <Policy>
      <PolicyName>OSS-Administrator</PolicyName>
      <PolicyType>Custom</PolicyType>
      <Description>OSS administrator permission</Description>
      <DefaultVersion>v1</DefaultVersion>
      <AttachDate>2015-01-23T12:33:18Z</AttachDate>
    </Policy>
    <Policy>
      <PolicyName>ReadOnlyAccess</PolicyName>
      <PolicyType>System</PolicyType>
      <Description>Read-only permission</Description>
      <DefaultVersion>v1</DefaultVersion>
      <AttachDate>2015-02-18T17:22:08Z</AttachDate>
    </Policy>
  </Policies>
</ListPoliciesForRoleResponse>
```

- JSON format

```
"RequestId": "7B8A4E7D-6CFF-471D-84DF-195A7A241ECB",
"Policies": {
  "Policy": [
    {
      "PolicyName": "OSS-Administrator",
      "PolicyType": "Custom",
      "Description": "OSS administrator permission",
      "DefaultVersion": "v1",
      "AttachDate": "2015-01-23T12:33:18Z"
    },
    {
      "PolicyName": "ReadOnlyAccess",
      "PolicyType": "System",
      "Description": "Read-only permission",
      "DefaultVersion": "v1",
      "AttachDate": "2015-02-18T17:22:08Z"
    }
  ]
}
```

1.8 安全设置接口

1.8.1 SetAccountAlias

Interface description

Sets the Alibaba Cloud account alias.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Operation interface, required. The parameter value is “SetAccountAlias.”

AccountAlias

- Type: String
- Required: Yes
- Description: Alibaba Cloud account alias. It consists of 3 to 32 characters.
- Restriction:

```
^[a-z0-9](([a-z0-9]|-(?!-))*[a-z0-9])?
```

Return parameters

Only public parameters are returned. For details, see [公共参数](#)

Required permissions

Action

```
ram:SetAccountAlias
```

Resource

```
acs:ram:*:${AccountId}:*
```

Error messages

InvalidParameter.AccountAlias.Format

- HTTP Status: 400
- Error Message: The format of parameter - “AccountAlias” is incorrect.

InvalidParameter.AccountAlias.Length

- HTTP Status: 400
- Error Message: The parameter - “AccountAlias” MUST be in the range of 3 and 32.

EntityAlreadyExists.AccountAlias

- HTTP Status: 409
- Error Message: The account alias does already exist.

Operation examples**Request example**

```
https://ram.aliyuncs.com/?Action=SetAccountAlias
&AccountAlias=myalias
&<Public request parameters>
```

Return example

- XML format

```
<SetAccountAliasResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
</SetAccountAliasResponse>
```

- JSON format

```
"RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368"
```

1.8.2 GetAccountAlias

Interface description

Views the Alibaba Cloud account alias.

Request parameters**Action**

- Type: String
- Required: Yes
- Description: Operation interface, required. The parameter value is “GetAccountAlias.”

Return parameters**AccountAlias**

- Type: String
- Description: Alibaba Cloud account alias.

Required permissions

Action

```
ram:GetAccountAlias
```

Resource

```
acs:ram:*:${AccountId}:*
```

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=GetAccountAlias  
&<Public request parameters>
```

Return example

- XML format

```
<GetAccountAliasResponse>  
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>  
  <AccountAlias>myalias</AccountAlias>  
</GetAccountAliasResponse>
```

- JSON format

```
"RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",  
"AccountAlias": "myalias"
```

1.8.3 ClearAccountAlias

Interface description

Clears the Alibaba Cloud account alias.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Operation interface, required. The parameter value is “ClearAccountAlias”.

Return parameters

Only public parameters are returned. For details, see [公共参数](#)

Required permissions

Action

```
ram:ClearAccountAlias
```

Resource

```
acs:ram:*:${AccountId}:*
```

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=ClearAccountAlias
&<Public request parameters>
```

Return example

- XML format

```
<ClearAccountAliasResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
</ClearAccountAliasResponse>
```

- JSON format

```
"RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368"
```

1.8.4 SetPasswordPolicy

Interface description

Sets the password policy for a user, including the password strength.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Operation interface, required. The parameter value is “SetPasswordPolicy”.

MinimumPasswordLength

- Type: Integer
- Required: No
- Description: Minimum password length.

RequireLowercaseCharacters

- Type: Boolean
- Required: No
- Description: Lowercase letters must be included.

RequireUppercaseCharacters

- Type: Boolean
- Required: No
- Description: Uppercase letters must be included.

RequireNumbers

- Type: Boolean
- Required: No
- Description: Numbers must be included.

RequireSymbols

- Type: Boolean
- Required: No
- Description: Special letters must be included.

Return parameters**PasswordPolicy**

- Type: [PasswordPolicy](#)
- Description: Password policy.

Required permissions

Action

```
ram:SetPasswordPolicy
```

Resource

```
acs:ram:*:${AccountId}:*
```

Error messages

InvalidParameter.MinimumPasswordLength.Range

- HTTP Status: 400
- Error Message: The parameter - "MinimumPasswordLength" MUST be in the range of 8 and 32

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=SetPasswordPolicy
&MinimumPasswordLength=12
&RequireLowercaseCharacters=true
&RequireUppercaseCharacters=true
&RequireNumbers=true
&RequireSymbols=true
&<Public request parameters>
```

Return example

- XML format

```
<SetPasswordPolicyResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <PasswordPolicy>
    <MinimumPasswordLength>12</MinimumPasswordLength>
    <RequireLowercaseCharacters>true</RequireLowercaseCharacters>
  >
    <RequireUppercaseCharacters>true</RequireUppercaseCharacters>
  >
    <RequireNumbers>true</RequireNumbers>
    <RequireSymbols>true</RequireSymbols>
  </PasswordPolicy>
</SetPasswordPolicyResponse>
```

- JSON format

```
"RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
"PasswordPolicy": {
  "MinimumPasswordLength": 12,
  "RequireLowercaseCharacters": true,
```

```
    "RequireUppercaseCharacters": true,  
    "RequireNumbers": true,  
    "RequireSymbols": true  
}
```

1.8.5 GetPasswordPolicy

Interface description

Obtains the password policy of a user, including the password strength.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Operation interface, required. The parameter value is “GetPasswordPolicy”.

Return parameters

PasswordPolicy

- Type: [PasswordPolicy](#)
- Description: Password policy.

Required permissions

Action

```
ram:GetPasswordPolicy
```

Resource

```
acs:ram:*:${AccountId}:*
```

Operation examples

Request example

```
https://ram.aliyuncs.com/?Action=GetPasswordPolicy  
&AccountAlias=myalias  
&<Public request parameters>
```

Return example

- XML format

```
<GetPasswordPolicyResponse>  
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>  
  <PasswordPolicy>
```

```
<MinimumPasswordLength>12</MinimumPasswordLength>
<RequireLowercaseCharacters>true</RequireLowercaseCharacters>
>
<RequireUppercaseCharacters>true</RequireUppercaseCharacters>
>
<RequireNumbers>true</RequireNumbers>
<RequireSymbols>true</RequireSymbols>
</PasswordPolicy>
</GetPasswordPolicyResponse>
```

- JSON format

```
"RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
"PasswordPolicy": {
  "MinimumPasswordLength": 12,
  "RequireLowercaseCharacters": true,
  "RequireUppercaseCharacters": true,
  "RequireNumbers": true,
  "RequireSymbols": true
}
```

1.8.6 SetSecurityPreference

Interface description

Set global security preferences

Request parameters

Action

- Type: String
- Required: Yes
- Description: Operation interface. System required parameter. Value: SetSecurityPreference.

EnableSaveMFATicket

- Type: Boolean
- Required: No
- Description: Whether to allow users to save the MFA authentication ticket at login. The ticket validity period is currently seven days.

AllowUserToChangePassword

- Type: Boolean
- Required: false
- Default: true
- Description: Whether to allow users to modify their own passwords

AllowUserToManageAccessKeys

- Type: Boolean
- Required: false
- Default: false
- Description: whether to allow users to manage their own AccessKeys

AllowUserToManageMFADevices

- Type: Boolean
- Required: false
- Default: true
- Description: whether to allow users to bind or unbind their own MFA devices

Response parameter**SecurityPreference**

- Type: [SetSecurityPreference](#)
- Description: Setting security preference

Required permissions**Action**

```
ram:SetSecurityPreference
```

Resource

```
acs:ram:*:${AccountId}:*
```

Example operation**Operation example**

```
https://ram.aliyuncs.com/?Action=SetSecurityPreference
&EnableSaveMFATicket=true
&AllowUserToChangePassword=true
&AllowUserToManageAccessKeys=false
&<Common Request Parameter>
```

Response example

- XML format

```
<SetSecurityPreferenceResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <SecurityPreference>
```

```

        <LoginProfilePreference>
            <EnableSaveMFATicket>true</EnableSaveMFATicket>
            <AllowUserToChangePassword>true</AllowUserToChangePas
sword>
        </LoginProfilePreference>LoginProfilePreference>
        <AccessKeyPreference>
            <AllowUserToManageAccessKeys>>false</AllowUserToManageAcc
essKeys>
        </AccessKeyPreference>
        <MFAPreference>
            <AllowUserToManageMFADevices>>false</AllowUserToManageMFA
Devices>
        </MFAPreference>
    </SecurityPreference>
</SetSecurityPreferenceResponse>

```

- JOSN format

```

"RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
"SecurityPreference": {
    "LoginProfilePreference": {
        "EnableSaveMFATicket": true,
        "AllowUserToChangePassword": true

    "AccessKeyPreference": {
        "AllowUserToManageAccessKeys": false

    "MFAPreference": {
        "AllowUserToManageMFADevices": true
    }
}

```

1.9 数据类型

1.9.1 User

描述

用户信息

节点名称

User

子节点

UserId

- 类型：String
- 描述：用户唯一标识

UserName

- 类型：String

- 描述：用户名称

CreateDate

- 类型：String
- 描述：创建时间

LastLoginDate

- 类型：String
- 描述：上次使用密码登录时间

1.9.2 LoginProfile

描述

用户登录配置

节点名称

LoginProfile

子节点

UserName

- 类型：String
- 描述：用户名

PasswordResetRequired

- 类型：Boolean
- 描述：要求下次登录时重设密码

MFABindRequired

- 类型：Boolean
- 描述：要求必须绑定多因素认证设备

1.9.3 MFADevice

描述

多因素认证设备

节点名称

MFADevice

子节点

SerialNumber

- 类型：String
- 描述：设备序列号

1.9.4 VirtualMFADevice

描述

虚拟多因素认证设备

节点名称

VirtualMFADevice

子节点

SerialNumber

- 类型：String
- 描述：设备序列号

Base32StringSeed

- 类型：String
- 描述：多因素认证设备密钥

QRCodePNG

- 类型：String
- 描述：密钥二维码PNG，使用Base64编码

ActivateDate

- 类型：String
- 描述：激活日期

User

- 类型：User
- 描述：绑定用户的基本信息

1.9.5 AccessKey

描述

访问密钥

节点名称

AccessKey

子节点

AccessKeyId

- 类型：String
- 描述：访问密钥标识

AccessKeySecret

- 类型：String
- 描述：访问密钥

Status

- 类型：String
- 描述：状态，激活或禁用

CreateDate

- 类型：String
- 描述：创建时间

1.9.6 Group

描述

组信息类型

节点名称

Group

子节点

GroupName

- 类型：String

- 描述：组名称

Comments

- 名称：Comments
- 类型：String
- 描述：备注信息

1.9.7 Role

描述

角色

节点名称

Role

子节点

RoleId

- 类型：String
- 描述：角色ID

RoleName

- 类型：String
- 描述：角色名称

Arn

- 类型：String
- 描述：角色的资源描述符

Description

- 类型：String
- 描述：角色的文字描述

AssumeRolePolicyDocument

- 类型：String
- 描述：角色扮演授权策略

CreateDate

- 类型：String
- 描述：创建时间

UpdateDate

- 类型：String
- 描述：更新时间

1.9.8 Policy

描述

授权策略

节点名称

Policy

子节点**PolicyName**

- 类型：String
- 描述：授权策略名称

PolicyType

- 类型：String
- 描述：授权策略类型

Description

- 类型：String
- 描述：授权策略描述

DefaultVersion

- 类型：String
- 描述：默认版本

CreateDate

- 类型：String
- 描述：创建时间

UpdateDate

- 类型：String
- 描述：修改时间

AttachmentCount

- 类型：Integer
- 描述：引用次数

1.9.9 PolicyVersion

描述

授权策略版本

节点名称

PolicyVersion

子节点**VersionId**

- 类型：String
- 描述：授权策略标识

IsDefaultVersion

- 类型：Boolean
- 描述：是否默认版本

CreateDate

- 类型：String
- 描述：创建时间

PolicyDocument

- 类型：String
- 描述：授权策略内容

1.9.10 PasswordPolicy

描述

密码策略

节点名称

PasswordPolicy

子节点

MinimumPasswordLength

- 类型：Integer
- 描述：最小密码长度

RequireLowercaseCharacters

- 类型：Boolean
- 描述：必须包含小写字母

RequireUppercaseCharacters

- 类型：Boolean
- 描述：必须包含大写字母

RequireNumbers

- 类型：Boolean
- 描述：必须包含数字

RequireSymbols

- 类型：Boolean
- 描述：必须包含字符

1.9.11 SecurityPreference

描述

安全首选项

节点名称

SecurityPreference

子节点[LoginProfilePreference]

EnableSaveMFATicket

- 类型：Boolean
- 必须：否

- 描述：是否允许在用户在登录时保存MFA验证票据，目前票据有效期是七天

AllowUserToChangePassword

- 类型：Boolean
- 必须：false
- 描述：是否允许用户修改自己的密码

子节点 **[AccessKeyPreference]AllowUserToManageAccessKeys**

- 类型：Boolean
- 必须：false
- 描述：是否允许用户管理自己的AccessKey

子节点 **[MFAPreference]**

AllowUserToManageMFADevices

- 类型：Boolean
- 必须：false
- 描述：是否允许用户管理自己绑定或解绑MFA设备

1.10 错误代码表

HTTP Status 400

InvalidParameter

InvalidParameter.UserName.InvalidChars

- HTTP Status：400
- Error Message：The parameter - "UserName" contains invalid chars.

InvalidParameter.UserName.Length

- HTTP Status：400
- Error Message：The parameter - "UserName" beyond the length limit.

InvalidParameter.NewUserName.InvalidChars

- HTTP Status：400
- Error Message：The parameter - "NewUserName" contains invalid chars.

InvalidParameter.NewUserName.Length

- HTTP Status : 400
- Error Message : The parameter - "NewUserName" beyond the length limit.

InvalidParameter.DisplayName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "DisplayName" contains invalid chars.

InvalidParameter.DisplayName.Length

- HTTP Status : 400
- Error Message : The parameter - "DisplayName" beyond the length limit.

InvalidParameter.NewDisplayName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "NewDisplayName" contains invalid chars.

InvalidParameter.NewDisplayName.Length

- HTTP Status : 400
- Error Message : The parameter - "NewDisplayName" beyond the length limit.

InvalidParameter.MobilePhone.Format

- HTTP Status : 400
- Error Message : The format of the parameter - "MobilePhone" is incorrect.

InvalidParameter.NewMobilePhone.Format

- HTTP Status : 400
- Error Message : The format of the parameter - "NewMobilePhone" is incorrect.

InvalidParameter.Email.Format

- HTTP Status : 400
- Error Message : The format of the parameter - "Email" is incorrect.

InvalidParameter.NewEmail.Format

- HTTP Status : 400
- Error Message : The format of the parameter - "NewEmail" is incorrect.

InvalidParameter.Comments.Length

- HTTP Status : 400
- Error Message : The parameter - "Comments" beyond the length limit.

InvalidParameter.NewComments.Length

- HTTP Status : 400
- Error Message : The parameter - "NewComments" beyond the length limit.

InvalidParameter.Password.TooWeak

- HTTP Status : 400
- Error Message : The parameter - "Password" is not compliant with the password policy.

InvalidParameter.VirtualMFADeviceName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "VirtualMFADeviceName" contains invalid chars.

InvalidParameter.VirtualMFADeviceName.Length

- HTTP Status : 400
- Error Message : The parameter - "VirtualMFADeviceName" beyond the length limit.

InvalidParameter.GroupName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "GroupName" contains invalid chars.

InvalidParameter.GroupName.Length

- HTTP Status : 400
- Error Message : The parameter - "GroupName" beyond the length limit.

InvalidParameter.NewGroupName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "NewGroupName" contains invalid chars.

InvalidParameter.NewGroupName.Length

- HTTP Status : 400
- Error Message : The parameter - "NewGroupName" beyond the length limit.

InvalidParameter.PolicyType

- HTTP Status : 400
- Error Message : The parameter - "PolicyType" is incorrect.

InvalidParameter.PolicyName.InvalidChars

- HTTP Status : 400
- Error Message : The parameter - "PolicyNam" contains invalid chars.

InvalidParameter.PolicyName.Length

- HTTP Status : 400
- Error Message : The parameter - "PolicyName" beyond the length limit.

InvalidParameter.PolicyDocument.Length

- HTTP Status : 400
- Error Message : The parameter - "PolicyName" beyond the length limit.

InvalidParameter.Description.Length

- HTTP Status : 400
- Error Message : The parameter - "Description" beyond the length limit.

InvalidParameter.VersionId.Format

- HTTP Status : 400
- Error Message : The format of the parameter - "VersionId" is incorrect.

InvalidParameter.AccountAlias.Format

- HTTP Status : 400
- Error Message : The format of parameter - "AccountAlias" is incorrect.

InvalidParameter.AccountAlias.Length

- HTTP Status : 400
- Error Message : The parameter - "AccountAlias" MUST be in the range of 3 and 63.

InvalidParameter.MinimumPasswordLength.Range

- HTTP Status : 400
- Error Message : The parameter - "MinimumPasswordLength" MUST be in the range of 8 and 32.

InvalidParameter.AuthenticationCode.Format

- HTTP Status : 400
- Error Message : The parameter - "AuthenticationCode" MUST be 6 numbers.

MalformedPolicyDocument

- HTTP Status : 400
- Error Message : {The error details}

HTTP Status 403**NoPermission**

- HTTP Status : 403
- Error Message : You are not authorized to do this action.

Inactive

- HTTP Status : 403
- Error Message : Your ram service is inactive, please activate it.

CheckAuthenticationCodeFail

- HTTP Status : 403
- Error Message : The parameter - "AuthenticationCode" is incorrect.

HTTP Status 404**EntityNotExistEntityNotExist.User**

- HTTP Status : 404
- Error Message : The user does not exist.

EntityNotExist.User.AccessKey

- HTTP Status : 404
- Error Message : The user access key does not exist.

EntityNotExist.User.LoginProfile

- HTTP Status : 404
- Error Message : The user login profile does not exist.

EntityNotExist.User.MFADevice

- HTTP Status : 404
- Error Message : The user has not bound any MFA device.

EntityNotExist.User.Policy

- HTTP Status : 404
- Error Message : The indicate policy of the user does not exist.

EntityNotExist.User.Group

- HTTP Status : 404
- Error Message : The user has not joined the group.

EntityNotExist.VirtualMFADevice

- HTTP Status : 404
- Error Message : The virtual mfa device does not exist.

EntityNotExist.Group

- HTTP Status : 404
- Error Message : The group does not exist.

EntityNotExist.Group.Policy

- HTTP Status : 404
- Error Message : The indicate policy attached to the group does not exist.

EntityNotExist.Policy

- HTTP Status : 404
- Error Message : The policy does not exist.

EntityNotExist.Policy.Version

- HTTP Status : 404
- Error Message : The policy version does not exist.

HTTP Status 409**EntityAlreadyExistsEntityAlreadyExists.User**

- HTTP Status : 409
- Error Message : The user does already EXIST.

EntityAlreadyExists.User.LoginProfile

- HTTP Status : 409
- Error Message : The user login profile does already EXIST.

EntityAlreadyExists.User.MFADevice

- HTTP Status : 409
- Error Message : The user has already bound a MFA device.

EntityAlreadyExists.User.Group

- HTTP Status : 409
- Error Message : The user has already joined the group.

EntityAlreadyExists.User.Policy

- HTTP Status : 409
- Error Message : The user has already been attached this policy.

EntityAlreadyExists.VirtualMFADevice

- HTTP Status : 409
- Error Message : The virtual mfa device does already EXIST.

EntityAlreadyExists.VirtualMFADevice.User

- HTTP Status : 409
- Error Message : The virtual mfa device does already been bound to a user.

EntityAlreadyExists.Group

- HTTP Status : 409
- Error Message : The group does already EXIST.

EntityAlreadyExists.Group.Policy

- HTTP Status : 409
- Error Message : The group has already been attached this policy.

EntityAlreadyExists.Policy

- HTTP Status : 409
- Error Message : The group does already EXIST.

EntityAlreadyExists.AccountAlias

- HTTP Status : 409
- Error Message : The account alias does already exist.

DeleteConflict.DeleteConflict.User.Group

- HTTP Status : 409
- Error Message : The user CAN NOT be in any group while deleting the user.

DeleteConflict.User.AccessKey

- HTTP Status : 409
- Error Message : The user CAN NOT has any access key while deleting the user.

DeleteConflict.User.LoginProfile

- HTTP Status : 409
- Error Message : The user CAN NOT has any login profile while deleting the user.

DeleteConflict.User.MFADevice

- HTTP Status : 409
- Error Message : The user CAN NOT has any mfa device while deleting the user.

DeleteConflict.User.Policy

- HTTP Status : 409
- Error Message : The user CAN NOT has any attached policy while deleting the user.

DeleteConflict.VirtualMFADevice.User

- HTTP Status : 409
- Error Message : The virtual mfa device CAN NOT been bound to any user while deleting it.

DeleteConflict.Group.User

- HTTP Status : 409
- Error Message : The group CAN NOT has any user member while deleting the group.

DeleteConflict.Group.Policy

- HTTP Status : 409
- Error Message : The entity CAN NOT has any attached policy while deleting the group.

DeleteConflict.Policy.User

- HTTP Status : 409
- Error Message : The policy CAN NOT been attached to any user while deleting the policy.

DeleteConflict.Policy.Group

- HTTP Status : 409
- Error Message : The policy CAN NOT been attached to any group while deleting the policy.

DeleteConflict.Policy.Version

- HTTP Status : 409
- Error Message : The policy CAN NOT has any version except the default version.

DeleteConflict.Policy.Version.Default

- HTTP Status : 409
- Error Message : The default policy version CAN NOT been deleted directly.

DeleteConflict.Role.Policy

- HTTP Status : 409
- Error Message : The role CAN NOT has any attached policy while deleting the role.

LimitExceededLimitExceeded.User

- HTTP Status : 409
- Error Message : The count of users beyond the current limits.

LimitExceeded.User.AccessKey

- HTTP Status : 409
- Error Message : The access key count of the user access keys beyond the current limits.

LimitExceeded.User.Group

- HTTP Status : 409
- Error Message : The count of groups the target user joined beyond the current limits.

LimitExceeded.User.Policy

- HTTP Status : 409
- Error Message : The policy count of the user attached policies beyond the current limits.

LimitExceeded.VirtualMFADevice

- HTTP Status : 409
- Error Message : The count of virtual mfa devices beyond the current limits.

LimitExceeded.Group

- HTTP Status : 409
- Error Message : The count of groups beyond the current limits.

LimitExceeded.Group.Policy

- HTTP Status : 409
- Error Message : The policy count of the group beyond the current limits.

LimitExceeded.Policy

- HTTP Status : 409
- Error Message : The count of policy beyond the current limits.

LimitExceeded.Policy.Version

- HTTP Status : 409
- Error Message : The count of policy version beyond the current limits.

LimitExceeded.Role

- HTTP Status : 409
- Error Message : The count of roles beyond the current limits.

LimitExceeded.Role.Policy

- HTTP Status : 409
- Error Message : The count of policies attached to the role beyond the current limits.

HTTP Status 500**InternalError**

- HTTP Status : 500
- Error Message : An internal error occurred.

1.11 关于RAM限制

请参考RAM API文档的[业务限制](#)。

2 API 参考 (STS)

2.1 简介

STS介绍

阿里云STS (Security Token Service) 是为阿里云账号 (或RAM用户) 提供短期访问权限管理的云服务。通过STS, 您可以为联盟用户 (您的本地账号系统所管理的用户) 颁发一个自定义时效和访问权限的访问凭证。联盟用户可以使用STS短期访问凭证直接调用阿里云服务API, 或登录阿里云管理控制台操作被授权访问的资源。

访问点

STS的默认访问点地址是: <https://sts.aliyuncs.com> , 用户必须使用https接入访问点。

术语表

术语	中文	说明
Federated identity	联盟用户身份	联盟用户的身份认证由客户自己管理
Policy	访问策略	用来描述授权策略的一种描述语言
Grantor	授权者	授权令牌的颁发者(云账号或RAM用户)
Name	被授权者	授权令牌的使用者(即联盟用户)

2.2 调用方式

2.2.1 请求结构

服务地址

STS服务地址请参考[服务地址](#)。

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。

HTTP请求方法

支持HTTP GET/POST方法发送请求，这种方式下请求参数需要包含在请求的URL中。(GET请求最大不得超过4KB, POST请求最大不得超过10MB)。

请求参数

每个请求都需要指定要执行的操作，即Action参数（例如AddUser），以及每个操作接口都需要包含的公共请求参数和指定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。

2.2.2 公共参数

Format

- 名称: Format
- 类型: String
- 必须: 否
- 说明: 返回值的类型，支持JSON与XML，默认为XML。

Version

- 名称: Version
- 类型: String
- 必须: 是
- 说明: API版本号，为日期形式：YYYY-MM-DD，本版本对应为2015-04-01。

AccessKeyId

- 名称: AccessKeyId
- 类型: String
- 必须: 是
- 说明: 阿里云颁发给用户的访问服务所用的密钥ID。

Signature

- 名称: Signature
- 类型: String

- 必须: 是
- 说明: 签名结果串, 关于签名的计算方法, 请参见签名机制。

SignatureMethod

- 名称: SignatureMethod
- 类型: String
- 必须: 是
- 说明: 签名方式, 目前仅支持HMAC-SHA1。

SignatureVersion

- 名称: SignatureVersion
- 类型: String
- 必须: 是
- 说明: 签名算法版本, 目前版本是1.0

SignatureNonce

- 名称: SignatureNonce
- 类型: String
- 必须: 是
- 说明: 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。

Timestamp

- 名称: Timestamp
- 类型: String
- 必须: 是
- 说明: 请求的时间戳。日期格式按照ISO8601标准表示, 并需要使用UTC时间。格式为: YYYY-MM-DDThh:mm:ssZ。例如, 2013-01-10T12:00:00Z (为北京时间2013年1月10日20点0分0秒)

2.2.3 返回结果

调用API服务后返回数据采用统一格式, 返回的HTTP状态码为2xx, 代表调用成功; 返回4xx或5xx的HTTP状态码代表调用失败。调用成功返回的数据格式主要有XML和JSON两种, 外部系统可以

在请求时传入参数来制定返回的数据格式，默认为XML格式。本文档中的返回示例为了便于用户查看，做了格式化处理，实际返回结果是没有进行换行、缩进等处理的。

成功结果

- **XML示例**

XML返回结果包括请求是否成功信息和具体的业务数据。示例如下：

```
<?xml version="1.0" encoding="utf-8"?>
<!--结果的根结点-->
<接口名称+Response>
  <!--返回请求标签-->
  <RequestId>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

调用接口出错后，将不会返回结果数据。调用方可根据附表<错误代码表>来定位错误原因。

当调用出错时，HTTP请求返回一个4xx或5xx的HTTP状态码。返回的消息体中是具体的错误代码及错误信息。另外还包含一个全局唯一的请求ID：RequestId和一个您该次请求访问的站点ID：HostId。在调用方找不到错误原因时，可以联系阿里云客服，并提供该HostId和RequestId，以便我们尽快帮您解决问题。

- **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."
}
```

2.2.4 签名

STS服务会对每个访问的请求进行身份验证，所以无论使用HTTP还是HTTPS协议提交请求，都需要在请求中包含签名（Signature）信息。STS通过使用**Access Key ID**和**Access Key Secret**进行对称加密的方法来验证请求的发送者身份。**Access Key ID**和**Access Key Secret**由阿里云官方颁发给访问者（可以通过阿里云官方网站申请和管理），其中**Access Key ID**用于标识访问者的身份；**Access Key Secret**是用于加密签名字符串和服务器端验证签名字符串的密钥，必须严格保密，只有阿里云和用户知道。

签名步骤

1. 使用请求参数构造规范化的请求字符串（Canonicalized Query String）

1. 按照参数名称的字典顺序对请求中所有的请求参数（包括文档中描述的“公共请求参数”和给定了的请求接口的自定义参数，但不能包括“公共请求参数”中提到Signature参数本身）进行排序。



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）。

2. 对每个请求参数的名称和值进行编码。名称和值要使用UTF-8字符集进行URL编码，URL编码的编码规则是：

- 对于字符 A-Z、a-z、0-9以及字符“-”、“_”、“.”、“~”不编码；
- 对于其他字符编码成“%XY”的格式，其中XY是字符对应ASCII码的16进制表示。比如英文的双引号（"）对应的编码就是%22
- 需要说明的是英文空格（ ）要被编码是%20，而不是加号（+）。



说明：

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

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

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

其中HTTPMethod是提交请求用的HTTP方法，比GET。percentEncode("/")是按照1.b中描述的URL编码规则对字符"/"进行编码得到的值，即"%2F"。

percentEncode(CanonicalizedQueryString)是对第1步中构造的规范化请求字符串按1.b中描述的URL编码规则编码后得到的字符串。

3. 按照[RFC2104](#)的定义，使用上面的用于签名的字符串计算签名HMAC值。注意：计算签名时使用的Key就是用户持有的Access Key Secret并加上一个"&"字符(ASCII:38)，使用的哈希算法是SHA1。
4. 按照Base64编码规则把上面的HMAC值编码成字符串，即得到签名值 (Signature) 。
5. 将得到的签名值作为Signature参数添加到请求参数中，即完成对请求签名的过程。注意：得到的签名值在作为最后的请求参数值提交给STS服务器的时候，要和其他参数一样，按照RFC3986的规则进行URL编码) 。

示例

以AssumeRole为例，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是：

```
GET&%2F&AccessKeyId%3Dtestid%26Action%3DAssumeRole%26Format%3DJSON%
26RoleArn%3Dacs%253Aram%253A%253A1234567890123%253Arole%252Ffirstrole
%26RoleSessionName%3Dclient%26SignatureMethod%3DHMAC-SHA1%26Signatur
```

```
eNonce%3D571f8fb8-506e-11e5-8e12-b8e8563dc8d2%26SignatureVersion%3D1.0%26Timestamp%3D2015-09-01T05%253A57%253A34Z%26Version%3D2015-04-01
```

假如使用的Access Key Id是“testid”，Access Key Secret是“testsecret”，用于计算HMAC的Key就是“testsecret&”，则计算得到的签名值是：

```
gNI7b0AyKZHxDgjBGPdgJ1Ce3L4=
```

签名后的请求URL为（注意增加了Signature参数）：

```
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
```

2.3 操作接口

2.3.1 AssumeRole

接口描述

通过该接口，获取一个扮演该角色的临时身份。

请求参数

Action

- 类型：String
- 必须：是
- 描述：系统规定参数，取值：AssumeRole

RoleArn

- 类型：String
- 必须：是
- 描述：指定角色的全局资源描述符(Aliyun Resource Name，简称Arn)。每个角色都有一个唯一的全局资源描述符，规定格式为 `acs:ram::$accountId:role/$roleName`，一个样例：`acs:ram::1234567890123456:role/samplerole`。您可以在RAM控制台的角色管理页面RAM控制台的角色管理列表中，进入角色详情页可以查看一个角色的RoleArn。

RoleSessionName

- 类型：String
- 必须：是

- 描述：用户自定义参数。此参数用来区分不同的Token，可用于用户级别的访问审计。
- 格式：

```
^[a-zA-Z0-9\.\@\-\_]+$
```

2-32个字符

Policy

- 名称：Policy
- 类型：String
- 必须：否
- 描述：授权策略[Policy 语法结构](#)，Policy长度限制为1024字节；您可以通过此参数限制生成的Token的权限，不指定则返回的Token将拥有指定角色的所有权限。

DurationSeconds

- 名称：DurationSeconds
- 类型：Integer
- 必须：否
- 描述：指定的过期时间，单位为秒。过期时间范围：900 ~ 3600，默认值为3600。

返回参数

Credentials

- 类型：[Credentials](#)
- 描述：访问凭证

AssumedRoleUser

- 描述：角色扮演临时身份

操作示例

HTTP Request

```
https://sts.aliyuncs.com?Action=AssumeRole
&RoleArn=acs:ram::1234567890123456:role/adminrole
&RoleSessionName=alice
&DurationSeconds=3600
&Policy=<url_encoded_policy>
&<公共请求参数>
```

HTTP Response

- **XML格式**

```
<AssumeRoleResponse>
  <RequestId>6894B13B-6D71-4EF5-88FA-F32781734A7F</RequestId>
  <AssumedRoleUser>
    <arn>acs:sts::1234567890123456:assumed-role/AdminRole/alice
  </arn>
    <AssumedRoleUserId>344584339364951186:alice<AssumedRoleUserId>
  </AssumedRoleUser>
  <Credentials>
    <AccessKeyId>STS.L4aBSCSJVMuKg5U1vFDw</AccessKeyId>
    <AccessKeySecret>wyLTsmyPGP1ohvww8xYgB29d1GI8KMiH2pKCNZ9</AccessKeySecret>
    <SecurityToken>CAESrAIIARKAAShQquMnLilbvEcIx06wCoqJufs8
sWwieUxu45hS9AvKNEte8KRUIWjWJ6Y+YHAPgNwi7yfRecMFydL2uP0gBI7LDi
o0RkbYlmJfIXHM2nGBPdml7kYEOxmJp2aDhbvvwVYIyt/8iES/R6N208wQh0Pk2bu
+/9dvalp6wOHF4gkFGhhTVFMuTDRhQlNDU0pWTXVLZzVVMXZGRHciBTQzMjc0K
gVhbG1jZTCpnJjwySk6BlJzYU1ENUJuCgExGmkKBUFFsbG93Eh8KDEFjdG1vb
kVxdWFscxIGQWN0aw9uGgcKBW9zczoqEj8KD1Jlc291cmNlRXF1YWxzEghSZ
XNvdXJjZRoJCiFhY3M6b3NzOio6NDMyNzQ6c2FtcGx1Ym94L2FsaWNlLyo=</SecurityToken>
    <Expiration>2015-04-09T11:52:19Z</Expiration>
  </Credentials>
</AssumeRoleResponse>
```

- **JSON格式**

```
{
  "Credentials": {
    "AccessKeyId": "STS.L4aBSCSJVMuKg5U1vFDw",
    "AccessKeySecret": "wyLTsmyPGP1ohvww8xYgB29d1GI8KMiH2pKCNZ9",
    "Expiration": "2015-04-09T11:52:19Z",
    "SecurityToken": "CAESrAIIARKAAShQquMnLilbvEcIx06wCoqJufs8
sWwieUxu45hS9AvKNEte8KRUIWjWJ6Y+YHAPgNwi7yfRecMFydL2uP0gBI7LDi
o0RkbYlmJfIXHM2nGBPdml7kYEOxmJp2aDhbvvwVYIyt/8iES/R6N208wQh0Pk2bu
+/9dvalp6wOHF4gkFGhhTVFMuTDRhQlNDU0pWTXVLZzVVMXZGRHciBTQzMjc0K
gVhbG1jZTCpnJjwySk6BlJzYU1ENUJuCgExGmkKBUFFsbG93Eh8KDEFjdG1vb
kVxdWFscxIGQWN0aw9uGgcKBW9zczoqEj8KD1Jlc291cmNlRXF1YWxzEghSZ
XNvdXJjZRoJCiFhY3M6b3NzOio6NDMyNzQ6c2FtcGx1Ym94L2FsaWNlLyo="
  },
  "AssumedRoleUser": {
    "arn": "acs:sts::1234567890123456:assumed-role/AdminRole/alice",
    "AssumedRoleUserId": "344584339364951186:alice"
  },
  "RequestId": "6894B13B-6D71-4EF5-88FA-F32781734A7F"
}
```

2.3.2 GetCallerIdentity

接口描述

通过该接口，可以获取当前调用者的身份信息。

请求参数

Action

- 类型：String
- 必须：是
- 描述：系统规定参数，取值：GetCallerIdentity

返回参数

AccountId

- 类型：String
- 描述：当前调用者所属云帐号的数字ID

UserId

- 类型：String
- 描述：当前调用者的用户ID。如果当前调用者是云帐号，则返回值与AccountId相同。

Arn

- 类型：String
- 描述：当前调用者的阿里云资源描述符(Aliyun Resource Names)

操作示例

HTTP Request

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

HTTP Response

- **XML**格式

```
<GetCallerIdentityResponse>
  <RequestId>2C9BE469-4A35-44D5-9529-CAA280B11603</RequestId>
  <AccountId>1968132000123456</AccountId>
  <UserId>216959339000654321</UserId>
  <Arn>acs:ram::1968132000123456:user/admin</Arn>
</GetCallerIdentityResponse>
```

- **JSON**格式

```
{
  "RequestId": "2C9BE469-4A35-44D5-9529-CAA280B11603",
  "AccountId": "1968132000123456",
  "UserId": "216959339000654321",
  "Arn": "acs:ram::1968132000123456:user/admin"
```

```
}
```

2.4 数据类型

2.4.1 Credentials

描述

访问凭证

节点名称

Credentials

子节点

AccessKeyId

- 类型：String
- 描述：访问密钥标识

AccessKeySecret

- 类型：String
- 描述：访问密钥

SecurityToken

- 类型：String
- 描述：安全令牌

Expiration

- 类型：String
- 描述：失效时间

2.4.2 AssumedRoleUser

描述

通过扮演角色接口获取的临时身份

节点名称

AssumedRoleUser

子节点

Arn

- 类型：String
- 描述：该角色临时身份的资源描述符

AssumedRoleUserId

- 类型：String
- 描述：该角色临时身份的用户ID

2.5 附录

2.5.1 错误代码表

HTTP Status 400

InvalidParameterInvalidParameter.RoleArn

- HTTP Status: 400
- ErrorMessage: The parameter RoleArn is wrongly formed.

InvalidParameter.RoleSessionName

- HTTP Status: 400
- ErrorMessage: The parameter RoleSessionName is wrongly formed.

InvalidParameter.DurationSeconds

- HTTP Status: 400
- ErrorMessage: The Min/Max value of DurationSeconds is 15min/1hr.

InvalidParameter.PolicyGrammar

- HTTP Status: 400
- ErrorMessage: The parameter Policy has not passed grammar check.

InvalidParameter.PolicySize

- HTTP Status: 400
- ErrorMessage: The size of Policy must be smaller than 1024 bytes.

HTTP Status 403

NoPermission

- HTTP Status: 403
- ErrorMessage: You are not authorized to do this action. You should be authorized by RAM.

HTTP Status 500

InternalError

- HTTP Status: 500
- ErrorMessage: STS Server Internal Error happened.

2.5.2 服务地址

服务地址选择建议

- 每个服务地址功能完全一样，请尽量同Region调用；
- Endpoint类型
 - 公网：互联网访问地址；
 - VPC：可在同Region内VPC中访问，无需开放公网访问权限；

服务地址

表 2-1: 国内中心

区域	域名
国内中心	公网:sts.aliyuncs.com

表 2-2: 大中华区

区域	RegionId	域名
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

区域	RegionId	域名
		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
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
香港	cn-hongkong	- 公网: sts.cn-hongkong.aliyuncs.com - VPC: sts-vpc.cn-hongkong.aliyuncs.com

表 2-3: 亚太

区域	RegionId	域名
亚太东南 1 (新加坡)	ap-southeast-1	公网: sts.ap-southeast-1.aliyuncs.com
亚太东南 2 (悉尼)	ap-southeast-2	公网: sts.ap-southeast-2.aliyuncs.com
亚太东南 3 (吉隆坡)	ap-southeast-3	公网: sts.ap-southeast-3.aliyuncs.com

区域	RegionId	域名
亚太东南 5 (雅加达)	ap-southeast-5	公网:sts.ap-southeast-5.aliyuncs.com
亚太东北 1 (东京)	ap-northeast-1	公网:sts.ap-northeast-1.aliyuncs.com
亚太南部 1 (孟买)	ap-south-1	公网:sts.ap-south-1.aliyuncs.com

表 2-4: 美洲-欧洲

区域	RegionId	域名
美国西部 1 (硅谷)	us-west-1	公网:sts.us-west-1.aliyuncs.com
美国东部 1 (弗吉尼亚)	us-east-1	公网:sts.us-east-1.aliyuncs.com
欧洲中部 1 (法兰克福)	eu-central-1	公网:sts.eu-central-1.aliyuncs.com
中东东部 1 (迪拜)	me-east-1	公网:sts.me-east-1.aliyuncs.com