Alibaba Cloud
Resource Access Management

API Reference

Issue: 20190212

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. **You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.**
2. **No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.**
3. **The content of this document may be changed due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.**
4. **This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity , applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use**

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

1 API overview

This topic lists all RAM APIs. For more information about OpenAPI resources, see *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
ListUsers	Used to list all RAM users
CreateLoginProfile	Used to enable console logon for a RAM user
GetLoginProfile	Used to view the logon configurations of a RAM user
DeleteLoginProfile	Used to disable console logon for a RAM user
UpdateLoginProfile	Used to modify user logon configurations
CreateAccessKey	Used to create an AccessKey (AK) for a RAM user
UpdateAccessKey	Used to change the AK status of a RAM user
DeleteAccessKey	Used to delete the AK of a RAM user
ListAccessKeys	Used to list the AK of s specified user
CreateVirtualMFADevice	Used to create a virtual MFA (VMFA) device
ListVirtualMFADevices	Used to list VMFA devices
DeleteVirtualMFADevice	Used to delete VMFA 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

API	Description
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
ListGroupsForUser	Used to list 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 a policy
GetPolicy	Used to obtain information about a specified policy

API	Description
DeletePolicy	Used to delete a specified policy
ListPolicies	Used to list policies
CreatePolicyVersion	Used to create a new policy version
GetPolicyVersion	Used to obtain the version of a policy
DeletePolicyVersion	Used to delete a version of a specified policy
ListPolicyVersions	Used to list all versions of a policy
SetDefaultPolicyVersion	Used to set the default version of a policy
AttachPolicyToUser	Used to add a policy to a specified user
DetachPolicyFromGroup	Used to remove a policy from a user
AttachPolicyToGroup	Used to add a policy to a specified user group
DetachPolicyFromRole	Used to remove a policy from a user group
AttachPolicyToRole	Used to add a policy to a specified role
DetachPolicyFromRole	Used to remove a policy from a role
ListEntitiesForPolicy	Used to list the entities using a policy
ListPoliciesForUser	Used to list the policies granted to a specified user
ListPoliciesForGroup	Used to list the policies granted to a specified user group
ListPoliciesForRole	Used to list the policies granted to a specified role

Security setting APIs

API	Description
SetAccountAlias	Used to set a cloud account alias
GetAccountAlias	Used to view a cloud account alias
ClearAccountAlias	Used to delete a cloud account alias
SetPasswordPolicy	Used to set the password policy for a user , including the password strength

API	Description
GetPasswordPolicy	Used to obtain the password policy of a user, including the password strength
SetSecurityPreference	Used to set the global security preference s

Data type

Type	Description
User	User information
LoginProfile	User logon configurations
MFADevice	MFA device
VirtualMFADevice	VMFA device
AccessKey	AK
Group	Group information
Role	Role
Policy	Policy
PolicyVersion	Policy version
PasswordPolicy	Password policy
SecurityPreference	Security preference

2 Introduction

2.1 RAM introduction

What is RAM?

Resource Access Management (RAM) is a resource access control service provided by Alibaba Cloud that enables you to centrally manage your users, systems, and applications, and control access to your resources.

Issues that are resolved by RAM

If you have purchased some cloud resources and several users in your organization need to use them, the users require access to your Alibaba Cloud account. In this case, the following issues may occur:

1. Sharing the AccessKey (AK) of your Alibaba Cloud account may mistakenly expose all of your cloud resources.
2. You cannot grant resource-specific access permissions to users.

Handling methods

- RAM allows you to create multiple RAM users under your Alibaba Cloud account and grant each RAM user the necessary resource operation permissions.
- RAM users cannot possess resources. They are centrally controlled and billed under your Alibaba Cloud account.
- You can create an independent password or AK for each RAM user, but RAM users do not have any operation permissions by default. RAM allows you to use access policies to control the authorization and access of your resources.

Functions

RAM provides the following functions:

- Centralized control of RAM users and their AKs: You can manage RAM users and their AKs, and attach and detach MFA devices to and from RAM users as needed.
- Centralized control of access permissions of RAM users: You can grant operation permissions for specific cloud resources to a RAM user.

- Centralized control of the resource access mode of RAM users: Through RAM, RAM users use secure channels (for example, SSL) to request access to specific cloud services at a specified time point in a specific network environment.
- Centralized control over cloud resources: You can centrally control the instances or data created by RAM users. When a user is removed from your organization, these instances or data will be retained.
- Centralized billing of charges: You need to pay for all charges incurred by resource operations of both your Alibaba Cloud account and RAM users.

Endpoint

The endpoint for API access is `https://ram.aliyuncs.com`.

Alibaba Cloud products supporting RAM

RAM can be integrated with a wide range of Alibaba Cloud products. For more information, see *Cloud services supporting RAM*.

Pricing

Alibaba Cloud RAM is provided free of charge. You only incur fees from RAM users using other services under your Alibaba Cloud account.

2.2 Terminology

Terminology	Description	Remarks
Account	Account	An account
User	User	A RAM user under an account
Group	Group or user group	A RAM user group under an account
Policy	Authorization policy	A language used to describe policies
AK (AccessKey)	Access key	An AK is composed of AccessKeyID and AccessKeySecret, and is used for identity authentication in Alibaba Cloud service API requests

Terminology	Description	Remarks
MFA (Multi-Factor Authentication)	Multi-Factor Authentication	You can associate your account or RAM users with MFA devices to enhance security.

3 Calling methods

3.1 Request Structure

Endpoint

The API access URL for RAM services is:

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

Communication protocol

To ensure communication security, RAM uses only the HTTPS secure channel to send requests.

HTTP request method

The system allows you to send HTTP GET/POST requests. In this method, request parameters must be included in the request URL.

Request parameters

You must use the Action parameter in each request to specify the operation to perform, and meanwhile you must add public parameters and interface service parameters to the request.

Character encoding

Requests and responses are encoded using UTF-8.

3.2 Public parameters

Format

- Name: Format
- Type: String
- Required: No
- Description: Return value type. JSON and XML are supported, and the default value is XML.

Version

- Name: Version

- Type: String
- Required: Yes
- Description: API version number. It is in the YYYY-MM-DD format and the current version number is 2015-05-01.

AccessKeyId

- Name: AccessKeyId
- Type: String
- Required: Yes
- Description: AccessKey (AK) ID

Signature

- Name: Signature
- Type: String
- Required: Yes
- Description: Message signature

SignatureMethod

- Name: SignatureMethod
- Type: String
- Required: Yes
- Description: Signature method. Currently, only HMAC-SHA1 is supported.

SignatureVersion

- Name: SignatureVersion
- Type: String
- Required: Yes
- Description: Signature algorithm version. The current version is 1.0.

SignatureNonce

- Name: SignatureNonce
- Type: String
- Required: Yes
- Description: A unique random number preventing network replay attacks. You must use different random numbers for different requests.

Timestamp

- **Name: Timestamp**
- **Type: String**
- **Required: Yes**
- **Description: Timestamp for a request. The date format follows the ISO8601 standard and uses UTC time. The timestamp format is:**

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

For example: 2013-01-10T12:00:00Z (equivalent to 2013-01-10 20:00:00 Beijing time)

Request example

```
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
          …
```

Public return parameters

Each time you send a request to call an interface, the system will return a unique identification code (RequestId) to you no matter the request is successful or not. This parameter is used to identify requests.

Example

- **XML example**

```
<? xml version="1.0" encoding="utf-8"? >
          <!—Result root node-->
          <Interface name+response>
          <!—Return request tag-->
          <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId
>
          <!—Return result data-->
          </Interface name+response>
```

- **JSON example**

```
          "RequestId": "4C467B38-3910-447D-87BC-AC049166F216"
          /* Return result data */
```

3.3 Return results

After the API service is called, data is returned in a unified format. The returned HTTP status code 2xx indicates that the call is successful, and 4xx or 5xx indicates that the call fails. For successful calls, the returned data are mainly in XML or JSON format. When a request is sent, an external system can enter a parameter to define the format of returned data, which is XML by default. In this topic, examples of returned results are formatted in a way that is easier for you to view. The actual return results are not formatted with line breaks or indentation.

Successful results

- XML example

  XML return results include a message stating whether the request is successful and the specific service data. For example:

```
<? xml version="1.0" encoding="utf-8"? >
<!—Result root node-->
<API name+response>
    <!—Return request tag-->
    <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
    <!—Return result data-->
</API name+response>
```

- JSON example

```
    "RequestId": "4C467B38-3910-447D-87BC-AC049166F216",
    /* Return result data */
```

Incorrect results

If there is an API call error, no result data is returned. You can locate error causes according to *Error code*.

When an error occurs in a call, an HTTP status code 4xx or 5xx will be returned for an HTTP request. The returned message body contains the specific error code and error message as well as the globally unique RequestId and the requested HostId. If you cannot locate the error cause, you can contact Alibaba Cloud customer service and provide your HostId and RequestId for quick troubleshooting.

- XML example

```
<? 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 example

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

3.4 Signature mechanism

RAM authenticates the identity of each access request. Therefore, no matter whether submitted through HTTP or HTTPS, a request must contain signature information. RAM uses AccessKeyId and AccessKeySecret for symmetric encryption to verify the identities of request senders. AccessKeyId and AccessKeySecret are officially issued to visitors by Alibaba Cloud (visitors can apply for and manage them on the official website of Alibaba Cloud). AccessKeyId indicates the identity of the visitor, and AccessKeySecret is the secret key used to encrypt the signature string and to verify the signature string on the server. It must be kept strictly confidential and should only be known only by Alibaba Cloud and the authenticated visitor.

Procedure

1. Use request parameters to construct a canonicalized query string.
   a. Sort all request parameters (including public request parameters and user-defined parameters with given request APIs described in this topic and excluding the Signature parameter) alphabetically by parameter name.

       Note:
      If you use the GET method to submit requests, these parameters will be included in the request URI, namely, the part after the question mark (?) following the ampersand (&) in the URI.

   b. Encode the name and value of each request parameter. URL encoding using the UTF-8 character set is required. URL encoding rules are as follows:

- Upper case letters from A to Z, lowercase letters from a to z, digits from 0 to 9 , and other characters including en dashes (-), underscores (_), periods (.), and tildes (~) are not encoded.
- Other characters are encoded in %XY format, with XY representing the characters' ASCII code in hexadecimal notation. For example, double quotation marks ("") are encoded as %22.
- It must be noted that spaces are encoded as %20 instead of plus signs (+).

 Note:
Generally, URL-encoded libraries (such as java.net.URLEncoder in Java) are encoded based on rules of the MIME type in application/x-www-form-urlencoded format. You can directly use this encoding method by replacing the plus sign (+) in the encoded string with %20 and the asterisk (*) with %2A. Also, you must change %7E back to the tilde (~) to conform to the encoding rules described above.

c. Connect the encoded parameter names and values with equal signs (=).
d. Connect the parameter name and value pairs connected by equal signs (=) alphabetically by parameter name with ampersands (&) to produce a canonicalized query string.

2. Use the canonicalized query string to construct the string for signature calculation according to the rule:

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

In this rule, HTTPMethod is the HTTP method (for example, GET) used for request submission. percentEncode("/") is the encoded value (namely, %2F) for the character "/" according to the URL encoding rules described in 1.b.

percentEncode(CanonicalizedQueryString) is the canonicalized query string encoded by following the URL encoding rules described in 1.b.

3. Use the string for signature calculation to calculate the HMAC value of the signature based on *RFC2104*. Note that the key used for signature calculation is your AccessKeySecret with an ampersand (&) (ASCII code: 38) and it is based on the hash algorithm SHA1.
4. Encode the HMAC value as a string according to the base64 encoding rules to obtain the signature.

5. Add the signature to the parameters of a request as a signature parameter. Note that the obtained signature value requires URL encoding based on RFC3986 rules like other parameters before it is submitted to the RAM server as the final request parameter value.

Example

Use CreateUser as an example. The request URL before signature is:

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

The corresponding StringToSign is:

```
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
```

Assume that the value of the AccessKeyId parameter is testid and that of the AccessKeySecret parameter is testsecret, and the key used for HMAC calculation is testsecret&. The calculated signature value is:

```
kRA2cnpJVacIhDMzXnoNZG9tDCI%3D
```

The signed request URL is (with the Signature parameter added):

```
https://ram.aliyuncs.com/?UserName=test&SignatureVersion=1.0&Format=
JSON&Timestamp=2015-08-18T03%3A15%3A45Z&AccessKeyId=testid&SignatureM
ethod=HMAC-SHA1&Version=2015-05-01&Signature=kRA2cnpJVacIhDMzXnoN
ZG9tDCI%3D&Action=CreateUser&SignatureNonce=6a6e0ca6-4557-11e5-86a2-
b8e8563dc8d2
```

4 User management APIs

4.1 CreateUser

Interface description

Creates a RAM user.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Operation name. It is a required parameter and its value is CreateUser .

UserName

- Type: String
- Required: Yes
- Description: User name, which can contain a maximum of 64 characters
- Format:

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

DisplayName

- Type: String
- Required: No
- Description: Display name, which can contain a maximum of 128 characters
- Format:

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

MobilePhone

- Type: String
- Required: No
- Description: A RAM user's mobile phone number

- Format: International area code-mobile phone number, for example, 86-18600008888

E-mail

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

Comments

- Type: String
- Required: No
- Description: Remarks, which can contain a maximum of 128 characters

Required permissions

Action

```
ram:CreateUser
```

Resource

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

Return parameters

User

- Type: *User*
- 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.

Examples

Request example

 **Note:**

To facilitate readability, parameters are not encoded in the following request examples (Percent Encode):

```
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 examples

- 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"
    }
}
```

4.2 GetUser

Interface description

Obtains detailed user information.

Request parameters

Action

- Name: Action
- Type: String
- Required: Yes
- Description: Operation name. It is a required parameter and its value is GetUser.

UserName

- Name: UserName
- Type: String
- Required: Yes
- Description: User name, for example, zhangqiang

- Format:

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

Return parameters

User

- Type: *User*
- Description: User information

Required permissions

Action

```
ram:GetUser
```

Resource

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

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.

Examples

Request example

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

Response examples

- XML format

```
<GetUserResponse>
```

```
    <RequestId>2D69A58F-345C-4FDE-88E4-BF5189484043</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>
        <UpdateDate>2015-02-11T03:15:21Z</UpdateDate>
        <LastLoginDate>2015-01-23T12:33:18Z</LastLoginDate>
    </User>
    <UserId>1227489245380721</UserId>
</GetUserResponse>
```

- **JSON format**

```
{
    "RequestId": "2D69A58F-345C-4FDE-88E4-BF5189484043",
    "User": {
        "UserId": "1227489245380721",
        "UserName": "zhangqiang",
        "Displayname": "Zhang Qiang ",
        "Maid phone": "86-18600008888 ",
        "Email": "zhangqiang@example.com",
        "Comments": "This is a cloud computing engineer ",
        "CreateDate": "2015-01-23T12:33:18Z",
        "UpdateDate": "2015-02-11T03:15:21Z",
        "LastLoginDate": "2015-01-23T12:33:18Z"
    }
}
```

4.3 ChangePassword

Interface description

Changes the password of a RAM user.

Request parameters

Action

- **Type: String**
- **Required: Yes**
- **Description: Operation name. It is a required parameter and its value is ChangePassword.**

OldPassword

- **Type: String**
- **Required: Yes**
- **Description: Old password**

NewPassword

- Type: String
- Required: Yes
- Description: Password, which must meet password strength requirements. For details about how to set the password strength, see *#unique_60*.

Return parameters

Only public parameters are returned. For details, see *Public parameters*.

Required permissions

Action

```
ram:ChangePassword
```

Resource

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

Error messages

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.

Examples

Request example

```
https://ram.aliyuncs.com/?Action=ChangePassword
&OldPassword=123456
```

```
&NewPassword=aw$2ad)d
&<Public request parameters>
```

Return examples

- XML format

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

- JSON format

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

5 Policy management APIs

5.1 DetachPolicyFromUser

Interface description

Detaches a specified policy from a user.

Request parameters

Action

- Type: String
- Required: Yes
- Description: Operation name. It is a required parameter and its value is DetachPolicyFromUser.

PolicyType

- Type: String
- Required: Yes
- Description: Policy type, which can be System or Custom

PolicyName

- Type: String
- Required: Yes
- Description: Policy name

UserName

- Type: String
- Required: Yes
- Description: User name, for example, zhangqiang

Return parameters

Only public parameters are returned. For details, see *Public parameters*.

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.**

Examples

Request example

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

Return examples

- **XML format**

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

- **JSON format**

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

6 Data types

6.1 User

Description

User information

Node name

User

Subnode

UserId

- **Type: String**
- **Description: Unique identifier of a user**

UserName

- **Type: String**
- **Description: User name**

CreateDate

- **Type: String**
- **Description: Creation time**

LastLoginDate

- **Type: String**
- **Description: Time of the latest password-based logon**

6.2 LoginProfile

Description

User login configurations

Node name

LoginProfile

Subnode

UserName

- **Type: String**
- **Description: User name**

PasswordResetRequired

- **Type: Boolean**
- **Description: The password must be reset when the next logon.**

MFABindRequired

- **Type: Boolean**
- **Description: An MFA device must be attached.**

6.3 MFADevice

Description

MFA device

Node name

MFADevice

Subnode

SerialNumber

- **Type: String**
- **Description: Device serial number**

6.4 VirtualMFADevice

Description

Virtual MFA (VMFA) device

Node name

VirtualMFADevice

Subnode

SerialNumber

- **Type: String**

- Description: Device serial number

Base32StringSeed

- Type: String
- Description: Key of an MFA device

QRCodePNG

- Type: String
- Description: Key QR code PNG using base64 encoding

ActivateDate

- Type: String
- Description: Activation date

User

- Type: User
- Description: Basic information about an attached user

6.5 AccessKey

Description

AccessKey (AK)

Node name

AccessKey

Subnode

AccessKeyId

- Type: String
- Description: AccessKey ID

AccessKeySecret

- Type: String
- Description: AK

Status

- Type: String
- Description: AK status, which can be Active or Inactive

CreateDate

- Type: String
- Description: Creation time

6.6 Group

Description

Group information types

Node name

Group

Subnode

GroupName

- Type: String
- Description: Group name

Comments

- Name: Comments
- Type: String
- Description: Remarks

6.7 Role

Description

Roles

Node name

Role

Subnode

RoleId

- Type: String
- Description: Role ID

RoleName

- Type: String

- Description: Role name

Arn

- Type: String
- Description: Role resource descriptor

Description

- Type: String
- Description: Role description

AssumeRolePolicyDocument

- Type: String
- Description: Policy for assuming a role

CreateDate

- Type: String
- Description: Creation time

UpdateDate

- Type: String
- Description: Update time

6.8 Policy

Description

Polic

Node name

Policy

Subnode

PolicyName

- Type: String
- Description: Policy name

PolicyType

- Type: String
- Description: Policy type

Description

- **Type: String**
- **Description: Policy description**

DefaultVersion

- **Type: String**
- **Description: Default policy version**

CreateDate

- **Type: String**
- **Description: Creation time**

UpdateDate

- **Type: String**
- **Description: Modification date**

AttachmentCount

- **Type: Integer**
- **Description: Referencing times**

6.9 PolicyVersion

Description

Policy versions

Node name

PolicyVersion

Subnode

VersionId

- **Type: String**
- **Description: Policy identifier**

IsDefaultVersion

- **Type: Boolean**
- **Description: Default version or not**

CreateDate

- Type: String
- Description: Creation time

PolicyDocument

- Type: String
- Description: Policy content

6.10 PasswordPolicy

Description

Password policy

Node name

PasswordPolicy

Subnode

MinimumPasswordLength

- Type: Integer
- Description: Minimum password length

RequireLowercaseCharacters

- Type: Boolean
- Description: Lowercase letters are mandatory.

RequireUppercaseCharacters

- Type: Boolean
- Description: Uppercase letters are mandatory.

RequireNumbers

- Type: Boolean
- Description: Numbers are mandatory.

RequireSymbols

- Type: Boolean
- Description: Characters are mandatory.

6.11 SecurityPreference

Description

Security preferences

Node name

SecurityPreference

Subnode [LoginProfilePreference]

EnableSaveMFATicket

- Type: Boolean
- Required: No
- Description: Whether users can save MFA credentials at logon. The credential validity period is seven days.

AllowUserToChangePassword

- Type: Boolean
- Required: No
- Description: Whether users can modify their own passwords

Subnode [AccessKeyPreference]

AllowUserToManageAccessKeys

- Type: Boolean
- Required: No
- Description: Whether users can manage their own AccessKeys (AKs)

Subnode [MFAPreference]

AllowUserToManageMFADevices

- Type: Boolean
- Required: No
- Description: Whether users can bind or unbind their own MFA devices