

Alibaba Cloud Resource Access Management

API 参考

Document Version 20190805

目次

1 呼び出し方式.....	1
1.1 リクエストの構造.....	1
1.2 パブリックパラメータ.....	1
1.3 署名のしくみ.....	4
1.4 返された結果の処理.....	6
2 API概要.....	8
3 ユーザー管理 API.....	12
3.1 ユーザーの作成 (CreateUser).....	12
3.2 ユーザー情報の取得 (GetUser).....	15
3.3 ユーザー情報の更新 (UpdateUser)	17
3.4 ユーザーの削除 (DeleteUser)	21
3.5 ユーザーの一覧表示 (ListUsers).....	23
3.6 ログインプロファイルの作成 (CreateLoginProfile)	26
3.7 ログインプロファイルの表示 (GetLoginProfile)	28
3.8 ログインプロファイルの削除 (DeleteLoginProfile)	30
3.9 ログインプロファイルの更新 (UpdateLoginProfile)	32
3.10 AccessKeyの作成 (CreateAccessKey)	34
3.11 アクセスキーの更新 (UpdateAccessKey).....	36
3.12 アクセスキーの削除 (DeleteAccessKey)	38
3.13 アクセスキーの一覧表示 (ListAccessKeys)	40
3.14 仮想MFAデバイスの作成 (CreateVirtualMFADevice)	42
3.15 仮想MFAデバイスの一覧表示 (ListVirtualMFADevices)	44
3.16 仮想MFAデバイスの削除 (DeleteVirtualMFADevice)	45
3.17 MFAデバイスのバインド (Bindmfadvice)	47
3.18 MFAデバイスのバインド解除 (UnbindMFADevice)	49
3.19 ユーザーMFA情報を取得 (GetUserMFAInfo)	51
3.20 パスワードの変更 (ChangePassword)	53
4 グループ管理インターフェイス.....	55
4.1 グループの作成 (CreatGroup)	55
4.2 グループ情報の取得 (GetGroup)	57
4.3 グループ情報の更新 (UpdateGroup).....	59
4.4 すべてのユーザーグループを一覧表示 (ListGroups)	61
4.5 グループの削除 (DeleteGroup)	64
4.6 ユーザーをユーザーグループに追加 (AddUserToGroup)	66
4.7 ユーザーグループからユーザーを削除 (RemoveUserFromGroup).....	68
4.8 ユーザーの所属グループ一覧を表示 (ListGropusForUser)	70
4.9 グループ内ユーザーの一覧表示 (ListUsersForGroup).....	72
5 ロール管理インターフェイス.....	76
5.1 ロールの作成 (CreateRole).....	76
5.2 ロールの取得 (GetRole).....	79

5.3	ロールの更新 (UpdateRole).....	81
5.4	ロールの一覧表示 (ListRoles).....	84
5.5	ロールの削除 (DeleteRole).....	86
6	権限付与ポリシー管理インターフェイス.....	88
6.1	権限付与ポリシーの作成 (CreatePolicy)	88
6.2	権限付与ポリシー情報の取得 (GetPolicy)	90
6.3	権限付与ポリシーの削除 (DeletePolicy)	92
6.4	権限付与ポリシーの一覧表示 (ListPolicies).....	94
6.5	権限付与ポリシーのバージョンを作成 (CreatePolicyVersion)	97
6.6	権限付与ポリシーのバージョンを取得 (GetPolicyVersion).....	99
6.7	権限付与ポリシーのバージョンを削除 (DeletePolicyVersion)	102
6.8	権限付与ポリシーのバージョンの一覧表示 (ListPolicyVersions).....	104
6.9	デフォルト権限付与ポリシーバージョンの設定 (SetDefaultPolicyVersion)	106
6.10	ユーザーに権限付与ポリシーをアタッチ (AttachPolicyToUser)	108
6.11	ユーザーから権限付与ポリシーをデタッチ (DetachPolicyFromUser).....	111
6.12	グループに権限付与ポリシーをアタッチ (AttachPolicyToGroup)	113
6.13	グループから権限付与ポリシーをデタッチ (DetachPolicyFromGroup).....	116
6.14	ロールに権限付与ポリシーをアタッチ (AttachPolicyToRole)	118
6.15	ロールからの権限付与ポリシーのデタッチ (DetachPolicyFromRole).....	121
6.16	ユーザーに付与された権限付与ポリシーの一覧表示 (ListPoliciesForUser).....	123
6.17	権限付与ポリシーを参照するエンティティの一覧表示 (ListEntitiesForPolicy)...	125
6.18	グループに付与された権限付与ポリシーの一覧表示 (ListPoliciesForGroup).....	129
6.19	ロールに付与された権限付与ポリシーの一覧表示 (ListPoliciesForRole)	131
7	セキュリティ管理 API.....	133
7.1	アカウントエイリアスの設定 (SetAccountAlias)	133
7.2	アカウントエイリアスの表示 (GetAccountAlias)	134
7.3	アカウントエイリアスの削除 (ClearAccountAlias)	135
7.4	パスワードポリシーの設定 (SetPasswordPolicy)	136
7.5	パスワードポリシーの取得 (GetPasswordPolicy)	139
7.6	グローバルセキュリティプリファレンスの設定 (SetSecurityPreference)	140

1 呼び出し方式

1.1 リクエストの構造

サービスアドレス

RAM サービスの API アクセス URL は次のとおりです：

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

通信プロトコル

通信のセキュリティを確保するために、RAM サービスは、安全な HTTPS チャンネルでのみリクエストを送信します。

HTTP リクエスト方式

HTTP GET/POST リクエストを送信できます。この方式では、リクエスト URL にリクエストパラメータを含める必要があります。

リクエストパラメータ

リクエストごとに Action パラメータを使用して、実行する操作を指定する必要があります。また、そのリクエストには、パブリックパラメータとインターフェイスサービスパラメータを追加する必要もあります。

文字エンコーディング

リクエストと返された結果は、UTF-8 文字セットでエンコードされます。

1.2 パブリックパラメータ

Format

- ・ 名称：Format
- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：戻り値の型。「JSON」と「XML」の2種類のタイプに対応しています。デフォルトでは「XML」です。

Version

- ・ 名称：Version

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：APIバージョン番号。形式：「YYYY-MM-DD」。現在の最新バージョンは「2015-05-01」です。

AccessKeyId

- ・ 名称：AccessKeyId
- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：アクセスキーID。

Signature

- ・ 名称：Signature
- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：メッセージの署名。

SignatureMethod

- ・ 名称：SignatureMethod
- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：署名方式。現在は、「HMAC-SHA1」のみに対応しています。

SignatureVersion

- ・ 名称：SignatureVersion
- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：署名アルゴリズムのバージョン。現在のバージョンは「1.0」です。

SignatureNonce

- ・ 名称：SignatureNonce
- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：一意の乱数で、ネットワーク反射攻撃を防ぎます。リクエストごとに異なる乱数を使用する必要があります。

Timestamp

- ・ 名称：Timestamp
- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：クエストのタイムスタンプ。日付形式はISO8601に従っており、UTC時刻を使用しています。タイムスタンプ形式の例は次の通りです：

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

例：2013-01-10T12:00:00Z (北京現地時間 1/10/2013 20:00:00と同じ)

リクエストの例

```
https :// ram . aliyuncs . com /
    Format = xml
    & Version = 2015 - 05 - 01
    & Signature = Pc5WB8gokV n0xfeu % 2FZV % 2BiNM1dgI % 3D
    & SignatureM ethod = HMAC - SHA1
    & SignatureN once = 1521552885 2396
    & SignatureV ersion = 1 . 0
    & AccessKeyI d = key - test
    & Timestamp = 2012 - 06 - 01T12 : 00 : 00Z
    ...
```

パブリックリターンパラメータ

インターフェイスを呼び出すリクエストを送信するたびに、成功したかどうかに関係なく、一意の識別コード「RequestId」が返されます。このパラメータは、各リクエストの特定に使用されます。

例

- ・ XML形式

```
<? xml version =" 1 . 0 " encoding =" utf - 8 "? >
    <!-- Result root node -->
    < Interface name + response >
    <!-- Return request tag -->
    < RequestId > 4C467B38 - 3910 - 447D - 87BC - AC049166F2
16 </ RequestId >
    <!-- Return result data -->
    </ Interface name + response >
```

- ・ JSON形式

```
" RequestId ": " 4C467B38 - 3910 - 447D - 87BC -
AC049166F2 16 "
/* Return result data */
```

1.3 署名のしくみ

STSサービスは、アクセスリクエストするたびにID認証を行います。それにより、HTTPまたはHTTPSを通じて送信する場合は、リクエストには署名情報が含まれる必要があります。RAMはAccessKey IDとAccess Key Secretを通じての対称暗号化という方式でリクエスト送信者の個人情報を認証します。AccessKey IDとAccessKeySecretはAlibabaクラウドよりリクエスト送信者に対して公式に発行されます（リクエスト送信者はAlibabaクラウドの公式Webサイトでこれらの情報を申請し、管理できます）。AccessKey IDはリクエスト送信者のIDを示します。AccessKey Secretは署名文字列を暗号化するために使用される秘密鍵であり、サーバーで署名文字列を認証する際にも用いられます。この情報は機密として厳密に取り扱い、Alibabaクラウドと認証済みリクエスト送信者以外には知られないようにしてください。

リクエスト署名プロセス

1. リクエストパラメータを使用して、正規化クエリ文字列を作成します。
 - a. すべてのリクエストパラメータ（「パブリックリクエストパラメーター」と、このドキュメントで取り上げる任意のリクエストインターフェイスに対するユーザー定義パラメータが含まれますが、「パブリックリクエストパラメータ」に説明のあるSignatureパラメータは含まれません）をパラメーター名のアルファベット順に並べ替えます。



注：

「GET」メソッドを使用してリクエストを送信する際に、これらのパラメータはリクエスト URL (URLのアンパーサンド「&」に続く疑問符「?」の後ろの部分) に含まれます。

- b. 各リクエストパラメータの名前と値をエンコードします。UTF-8 文字セットを使用した URL エンコードが必要です。URL エンコーディングのルールは次のとおりです。
 - ・ 大文字 (A ~ Z)、小文字 (a ~ z)、整数 (0 ~ 9)、および一部の記号 (ハイフン “-”、アンダーバー “_”、ピリオド “.”、チルダ “~” など) はエンコードされません。
 - ・ 他の文字は “%XY” 形式でエンコードします。この XY は、文字の ASCII コードを 16 進表記することを意味します。たとえば、二重引用符 (“ ”) は “%22” です。
 - ・ 半角スペース () はプラス記号 “+” ではなく、 “%20” としてエンコードされることに注意してください。



注：

一般的に、URL エンコーディングされたライブラリ (Java の「java.net.URLEncoder」など) は、「application/x-www-form-urlencoded」形式の MIME タイプのルールに基づいてエンコードされます。このエンコーディング方法を使用するには、前述のエンコーディングルールに合わせて、エンコードされた文字列内のプラス記号「+」を「%20」に、アスタリスク「*」を「%2A」に直接置き換え、「%7E」をチルダ「~」に戻します。

- c. エンコードしたパラメーター名と値を半角の等号「=」で連結します。
 - d. 等号で結んだパラメーター名と値のペアを、アンパーサンド「&」を使用してパラメーター名のアルファベット順に連結して、正規化クエリ文字列を作成します。
2. 次のルールに従って正規化クエリ文字列を使用し、署名計算用の文字列を作成します：

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

ここで「HTTPMethod」は、リクエストの送信に使用する HTTP メソッド (“GET” など) です。「percentEncode (“/”)」は、1.b で説明した URL エンコーディングルールに従って文字「/」をエンコードした値 (“%2F”) です。

「percentEncode(CanonicalizedQueryString)」は、1.b の URL エンコーディングルールに従ってエンコードした正規化クエリ文字列です。

3. [RFC2104](#) の定義に基づいて、署名計算のための文字列を使用して署名の HMAC 値を計算します。注意: 署名計算に使用するキーは、Access Key Secret にアンパーサンド “&” (ASCII: 38) を付加したもので、ハッシュアルゴリズム SHA1 に基づいています。
4. Base64 エンコーディングルールに基づいて HMAC 値を文字列にエンコードし、署名値を取得します。
5. 取得した署名値を「Signature」パラメータとしてリクエストパラメータに追加し、リクエスト署名プロセスを完了します。注意: 取得した署名値は、最終的なリクエストパラメーター値として STS サーバーに送信する前に、他のパラメーターと同様、RFC3986 ルールに基づいて URL エンコーディングする必要があります。

例

「CreateUser」を例として使用した場合、署名前のリクエスト URL は次のようになります：

```
https://ram.aliyuncs.com/?UserName=test&SignatureVersion=1.0&Format=JSON&Timestamp=2015-08-18T03:30:15Z&AccessKeyId=testid&SignatureMethod=HMAC-SHA1&
```

```
Version = 2015 - 05 - 01 & Action = CreateUser & SignatureNonce = 6a6e0ca6 - 4557 - 11e5 - 86a2 - b8e8563dc8 d2
```

対応する「StringToSign」は:

```
GET & % 2F & AccessKeyId % 3Dtestid % 26Action % 3DCreateUser % 26Format % 3DJSON % 26SignatureMethod % 3DHMAC - SHA1 % 26SignatureNonce % 3D6a6e0ca6 - 4557 - 11e5 - 86a2 - b8e8563dc8 d2 % 26SignatureVersion % 3D1 . 0 % 26Timestamp % 3D2015 - 08 - 18T03 % 253A15 % 253A45Z % 26Username % 3Dtest % 26Version % 3D2015 - 05 - 01
```

AccessKeyIdパラメータ値を「testid」、AccessKeySecretパラメータ値を「testsecret」とすると、HMACの計算に使用するKeyは「Testsecret&」、計算結果の署名値は次のようになります:

```
kRA2cnpJVa cIhDMzXnoN ZG9tDCI % 3D
```

署名付きリクエストURLは次のようになります(「Signature」パラメータが追加されています)。

```
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 = kRA2cnpJVa cIhDMzXnoN ZG9tDCI % 3D & Action = CreateUser & SignatureNonce = 6a6e0ca6 - 4557 - 11e5 - 86a2 - b8e8563dc8 d2
```

1.4 返された結果の処理

API サービスを呼び出した後、データは一様な形式で返されます。返された HTTP ステータスコードが 2xx という形式の場合、呼び出しは成功です。返された HTTP ステータスコードが 4xx または 5xx の場合、呼び出しは失敗です。呼び出しが成功した場合、データは主に XML と JSON の 2 つの形式で返されます。外部システムはリクエストの送信時に、パラメータで返されるデータの形式を指定できます。デフォルトではXML形式です。本ドキュメントでは、返される結果の例を見やすい形式で表示しています。実際の結果は、改行やインデントなどで体裁が整えられているわけではありません。

呼び出し成功の例

- ・ XML形式

XML 形式で返された結果には、リクエストが成功したかどうかを示すメッセージと、特定のサービスデータが含まれます。例:

```
<? xml version = " 1 . 0 " encoding = " utf - 8 " ? >
<!-- Result root node -->
< Interface name + response >
```

```

    <!-- Return request tag -->
    < RequestId > 4C467B38 - 3910 - 447D - 87BC - AC049166F2 16 </
RequestId >
    <!-- Return result data -->
</ Interface name + response >

```

- ・ JSON形式

```

    " RequestId ": " 4C467B38 - 3910 - 447D - 87BC - AC049166F2 16
",
    /* Return result data */

```

呼び出し失敗の例

インターフェイス呼び出しでエラーが発生した場合、結果が返されません。エラーの原因は、添付の付録の[#unique_6](#)で確認できます。

呼び出しでエラーが発生した場合、HTTP リクエストに対して 4xx または 5xx という形式の HTTP ステータスコードが返されます。返されるメッセージには、特定のエラーコードとエラーメッセージが含まれます。また、メッセージには、全体で一意的な「RequestId」と、リクエストした「HostId」も含まれています。エラーの原因を特定できない場合は、Alibaba Cloud カスタマーサービスにご連絡ください。その際、よりスムーズに問題を解決できるよう「HostId」と「RequestId」をご提示ください。

- ・ XML形式

```

<? xml version = " 1 . 0 " encoding = " UTF - 8 " ? >
< Error >
    < RequestId > 8906582E - 6722 - 409A - A6C4 - 0E7863B733 A5 </
RequestId >
    < HostId > ram . aliyuncs . com </ HostId >
    < Code > InvalidParameter </ Code >
    < Message > The specified parameter " Action or Version
" is not valid .</ Message >
</ Error >

```

- ・ JSON形式

```

    " RequestId ": " 7463B73D - 35CC - 4D19 - A010 - 6B8D65D242 EF
",
    " HostId ": " ram . aliyuncs . com ",
    " Code ": " InvalidParameter ",
    " Message ": " The specified parameter \" Action or
Version \" is not valid ."

```

2 API概要

本ページにはすべての利用可能なRAM APIが記述されています。OpenAPIリソースの詳細は、[API Explorer](#)を参照してください。

ユーザー管理API

API	説明
ユーザーの作成 (CreateUser)	RAMユーザーを作成します
ユーザー情報の取得 (GetUser)	ユーザーの詳細情報を取得します
ユーザー情報の更新 (UpdateUser)	基本ユーザー情報を更新します
ユーザーの削除 (DeleteUser)	RAMユーザーを削除します
ユーザーの一覧表示 (ListUsers)	すべてのRAMユーザーを表示します
ログインプロファイルの作成 (CreateLoginProfile)	RAMユーザーのWEBコンソールへのログインを有効にします
ログインプロファイルの表示 (GetLoginProfile)	RAMユーザーのログイン構成情報を確認します
ログインプロファイルの削除 (DeleteLoginProfile)	RAMユーザーのWEBコンソールへのログインを無効にします
ログインプロファイルの更新 (UpdateLoginProfile)	ユーザーのログイン構成情報を調整します
AccessKeyの作成 (CreateAccessKey)	RAMユーザーのためのアクセスキーを作成します
アクセスキーの更新 (UpdateAccessKey)	RAMユーザーのアクセスキーのステータスを変更します
アクセスキーの削除 (DeleteAccessKey)	RAMユーザーのアクセスキーを削除します
アクセスキーの一覧表示 (ListAccessKeys)	特定ユーザーのアクセスキーを表示します
仮想MFAデバイスの作成 (CreateVirtualMFADevice)	仮想MFAデバイスを作成します
仮想MFAデバイスの一覧表示 (ListVirtualMFADevices)	仮想MFAデバイスを一覧表示します
仮想MFAデバイスの削除 (DeleteVirtualMFADevice)	仮想MFAデバイスを削除します
MFAデバイスのバインド (Bindmfadvice)	MFAデバイスをバインドします
UnbindMFADevice	MFAデバイスをバインド解除します

API	説明
GetUserMFAInfo	特定RAMユーザーにバインドしたMFAデバイス情報を取得します
ChangePassword	RAMユーザーのパスワードを変更します

グループ管理API

API	説明
CreateGroup	ユーザーグループを作成します
GetGroup	ユーザーグループの情報を取得します
UpdateGroup	ユーザーグループの情報を更新します
ListGroup	すべてのユーザーグループを表示します
DeleteGroup	特定ユーザーグループを削除します
AddUserToGroup	特定ユーザーグループにRAMユーザーを追加します
RemoveUserFromGroup	ユーザーグループからRAMユーザーを削除します
ListGroupsForUser	特定RAMユーザーが参加しているグループの情報を表示します
ListUsersForGroup	特定ユーザーグループに参加しているRAMユーザーを表示します

ロール管理API

API	説明
ロールの作成 (CreateRole)	ロールを作成します
ロールの取得 (GetRole)	ロール情報を取得します
ロールの更新 (UpdateRole)	ロール情報を更新します
ロールの一覧表示 (ListRoles)	ロールを一覧表示します
ロールの削除 (DeleteRole)	特定ロールを削除します

権限付与ポリシー管理API

API	説明
権限付与ポリシーの作成 (CreatePolicy)	権限付与ポリシーを作成します
権限付与ポリシー情報の取得 (GetPolicy)	特定権限付与ポリシーの情報を取得します

API	説明
権限付与ポリシーの削除 (DeletePolicy)	特定権限付与ポリシーを削除します
権限付与ポリシーの一覧表示 (ListPolicies)	権限付与ポリシーを一覧表示します
権限付与ポリシーのバージョンを作成 (CreatePolicyVersion)	権限付与ポリシーバージョンを新規作成します
権限付与ポリシーのバージョンを取得 (GetPolicyVersion)	権限付与ポリシーのバージョン情報を取得します
権限付与ポリシーのバージョンを削除 (DeletePolicyVersion)	特定権限付与ポリシーのバージョンを削除します
権限付与ポリシーのバージョンの一覧表示 (ListPolicyVersions)	権限付与ポリシーのすべてのバージョンを一覧表示します
デフォルト権限付与ポリシーバージョンの設定 (SetDefaultPolicyVersion)	権限付与ポリシーバージョンをデフォルトバージョンに設定します
ユーザーに権限付与ポリシーをアタッチ (AttachPolicyToUser)	特定ユーザーに権限付与ポリシーをアタッチします
ユーザーから権限付与ポリシーをデタッチ (DetachPolicyFromUser)	ユーザーから権限付与ポリシーを削除します
グループに権限付与ポリシーをアタッチ (AttachPolicyToGroup)	特定ユーザーグループに権限付与ポリシーをアタッチします
グループから権限付与ポリシーをデタッチ (DetachPolicyFromGroup)	ユーザーグループから権限付与ポリシーを削除します
ロールに権限付与ポリシーをアタッチ (AttachPolicyToRole)	特定ロールに権限付与ポリシーをアタッチします
ロールからの権限付与ポリシーのデタッチ (DetachPolicyFromRole)	ロールから権限付与ポリシーを削除します
権限付与ポリシーを参照するエンティティの一覧表示 (ListEntitiesForPolicy)	権限付与ポリシーにアタッチされているエンティティを一覧表示します
ユーザーに付与された権限付与ポリシーの一覧表示 (ListPoliciesForUser)	特定ユーザーにアタッチされている権限付与ポリシーを一覧表示します
グループに付与された権限付与ポリシーの一覧表示 (ListPoliciesForGroup)	特定ユーザーグループにアタッチされている権限付与ポリシーを一覧表示します
ロールに付与された権限付与ポリシーの一覧表示 (ListPoliciesForRole)	特定ロールにアタッチされている権限付与ポリシーを一覧表示します

セキュリティ設定API

API	説明
アカウントエイリアスの設定 (SetAccountAlias)	クラウドアカウントエイリアスを設定します
アカウントエイリアスの表示 (GetAccountAlias)	クラウドアカウントエイリアスを確認します
アカウントエイリアスの削除 (ClearAccountAlias)	クラウドアカウントエイリアスを削除します
パスワードポリシーの設定 (SetPasswordPolicy)	ユーザーに強度を含めたパスワードポリシーを設定します。
パスワードポリシーの取得 (GetPasswordPolicy)	ユーザーに設定している強度を含めたパスワードポリシーを取得します
グローバルセキュリティプリファレンスの設定 (SetSecurityPreference)	グローバルセキュリティプリファレンスを設定します

データ型

データ型	説明
User	ユーザー情報
ログインプロファイル (LoginProfile)	ユーザーログイン構成情報
MFAデバイス (MFADevice)	MFAデバイス
仮想MFAデバイス (VirtualMFADevice)	仮想MFAデバイス
アクセスキー (AccessKey)	アクセスキー
グループ (Group)	グループ情報
ロール (Role)	ロール
権限付与ポリシー (Policy)	権限付与ポリシー
ポリシーバージョン (PolicyVersion)	権限付与ポリシーバージョン
パスワードポリシー (PasswordPolicy)	パスワードポリシー
セキュリティ設定 (SecurityPreference)	セキュリティプリファレンス

3 ユーザー管理 API

3.1 ユーザーの作成 (CreateUser)

インターフェイス説明

RAM ユーザーを作成します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス、必須。パラメータ値：「CreateUser」。

UserName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ユーザーの名前。最大文字数は64文字です。
- ・ 形式：

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

DisplayName

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：表示名。最大文字数は12文字です。
- ・ 形式：

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

MobilePhone

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：RAMユーザーの携帯電話番号。
- ・ 形式：<国際エリアコード>-<番号> (例: 86-18600008888)

E-mail

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：RAMユーザーのEmailアドレス。

Comments

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：補足情報。 最大文字数は128文字です。

必要な権限

Action

```
ram : CreateUser
```

Resource

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

返回参数

User

- ・ データ型：**User**型。
- ・ 説明：ユーザー情報。

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

InvalidParameter.DisplayName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「DisplayName」に無効な文字が含まれています。

InvalidParameter.DisplayName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「DisplayName」 は長さの制限を超えています。

InvalidParameter.Comments.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「Comments」 は長さの制限を超えています。

InvalidParameter.MobilePhone.Format

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「MobilePhone」 の形式は正しくありません。

InvalidParameter.Email.Format

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「Email」 の形式は正しくありません。

EntityAlreadyExists.User

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ユーザーは既に存在しています。

LimitExceeded.User

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ユーザーの数は制限を超えています。

操作の例

リクエストの例



注：

注意: 読みやすくするため、次のリクエストの例ではパラメータはエンコードされていません。

```
https :// ram . aliyuncs . com /? Action = CreateUser
& Username = zhangqiang
& DisplayNam e = zhangqiang
& MobilePhon e = 86 - 1868888888 8
& Email = zhangqiang @ example . com
& Comments = This is a cloud computing engineer .
&< Public request parameters >
```

戻り値の例

- ・ XML形式

```
< CreateUser Response >
< RequestId > 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 </
RequestId >
```

```
< User >
< UserId > 1227489245 380721 </ UserId >
< UserName > zhangqiang </ UserName >
< DisplayNam e > zhangqiang </ DisplayNam e >
< MobilePhon e > 86 - 1860000888 8 </ MobilePhon e >
< Email > zhangqiang@example.com </ Email >
< Comments > This is a cloud computing engineer .</
Comments >
< CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </ CreateDate >
</ User >
</ CreateUser Response >
```

・ JSON形式

```
{
  " RequestId ": " 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68
",
  " User ": {
    " UserId ": " 1227489245 380721 ",
    " UserName ": " zhangqiang ",
    " DisplayNam e ": " zhangqiang ",
    " MobilePhon e ": " 86 - 1860000888 8 ",
    " Email ": " maid ",
    " Comments ": " This is a cloud computing engineer
".
    " CreateDate ": " 2015 - 01 - 23T12 : 33 : 18Z "
  }
}
```

3.2 ユーザー情報の取得 (GetUser)

インターフェイス説明

詳細なユーザー情報を取得します。

リクエストパラメータ

Action

- ・ 名称 : Action
- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : 操作インターフェイス、必須。パラメータ値 : 「GetUser」。

Username

- ・ 名称 : Username
- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : ユーザーの名前。例 : zhangqiang

- ・ 形式：

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

リターンパラメータ

User

- ・ データ型：User型 **User**
- ・ 説明：ユーザー情報。

必要な権限

Action

```
ram : GetUser
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

```
<GetUserResponse>
```

```

    < RequestId > 2D69A58F - 345C - 4FDE - 88E4 - BF51894840 43 </
RequestId >
    < User >
        < UserId > 1227489245 380721 </ UserId >
        < UserName > zhangqiang </ UserName >
        < Displayname > zhangqiang </ displayname >
        < MobilePhone > 86 - 1860000888 8 </ 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 > 1227489245 380721 </ UserId >
</ GetUserResponse >

```

・ JSON形式

```

{
  "RequestId": "2D69A58F - 345C - 4FDE - 88E4 - BF51894840 43",
  "User": {
    "UserId": "1227489245 380721",
    "UserName": "zhangqiang",
    "DisplayName": "Zhang Qiang",
    "MobilePhone": "86 - 1860000888 8",
    "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"
  }
}

```

3.3 ユーザー情報の更新 (UpdateUser)

インターフェイス説明

基本的なユーザー情報を更新します。

リクエストパラメータ

Action

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 操作インターフェイス、必須。パラメータ値: 「UpdateUser」。

UserName

- ・ データ型: String
- ・ 必須項目: はい

- ・ 説明：ユーザーの名前。
- ・ 形式：

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

NewUserName

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：新しいユーザー名。
- ・ 形式：

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

NewDisplayName

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：新しい表示名。

NewMobilePhone

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：RAMユーザーの新しい携帯電話番号。
- ・ 形式：<国際エリアコード>-<番号> (例: 86-18600008888)

NewEmail

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：RAMユーザーの新しいEmailアドレス。

NewComments

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：新しい補足情報。 最大文字数は128文字です。

リターンパラメータ

User

- ・ データ型：User型 **User**

- ・ 説明：ユーザー情報。

必要な権限

Action

```
ram : UpdateUser
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

InvalidParameter.NewUserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「NewUserName」に無効な文字が含まれています。

InvalidParameter.NewUserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「NewUserName」は長さの制限を超えています。

InvalidParameter.NewDisplayName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「NewDisplayName」に無効な文字が含まれています。

InvalidParameter.NewDisplayName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「NewDisplayName」は長さの制限を超えています。

InvalidParameter.NewComments.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「NewComments」は長さの制限を超えています。

InvalidParameter.NewMobilePhone.Format

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「NewComments」の形式は正しくありません。

InvalidParameter.NewEmail.Format

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「NewEmail」の形式は正しくありません。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

Entityalreadyexists. User

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ユーザーは既に存在しています。

操作の例**リクエストの例**

```
https://ram.aliyuncs.com/?Action=UpdateUser
&UserName=zhangqiang
&NewUserName=xiaoqiang
&NewMobilePhone=86-18600008888
&NewEmail=zhangqiang@example.com
&<Public request parameters>
```

戻り値の例

- ・ XML形式

```
< CreateUser Response >
  < RequestId > 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 </
RequestId >
  < User >
    < UserId > 1227489245 380721 </ UserId >
    < UserName > xiaoqiang </ UserName >
    < DisplayName > zhangqiang </ DisplayName >
    < MobilePhone > 86 - 18600008888 8 </ 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 >
  </ User >
</ CreateUser Response >
```

- ・ XML形式

```
{
```



```
" RequestId ": " 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68
",
" User ": {
  " UserId ": " 1227489245 380721 ",
  " Username ": " Xiaoqiang ",
  < DisplayNam e > zhangqiang </ DisplayNam e >
  " MobilePhon e ": " 86 - 1860000888 8 ",
  " Email ": " zhangqiang @ example . com ",
  < Comments > This is a cloud computing engineer
.</ Comments >
  " CreateDate ": " 2015 - 01 - 23T12 : 33 : 18Z ",
  " UpdateDate ": " 2015 - 02 - 11T03 : 15 : 21Z "
}
}
```

3.4 ユーザーの削除 (DeleteUser)

インターフェイスの説明

RAMユーザーを削除します

ユーザーを削除する前に、権限付与ポリシーを所有しているか、ユーザーグループに所属しているか確かめてください。

リクエストパラメータ

Action

- ・ 名前: Action
- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 必須の システムパラメータ。その値は「DeleteUser」。

UserName

- ・ 名前: UserName
- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: ユーザー名。 例: zhangqiang。
- ・ 形式:

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

リターンパラメータ

パブリックパラメーターのみが返されます。 詳細について、次を参照してください。 [パブリックパラメータ](#)

必要な権限

Action

```
ram : DeleteUser
```

リソース

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード: 400
- ・ エラーメッセージ: パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード: 400
- ・ エラーメッセージ: パラメータ - 「UserName」が長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード: 404
- ・ エラーメッセージ: ユーザーが存在しません。

DeleteConflict.User.Group

- ・ HTTPステータスコード: 409
- ・ エラーメッセージ: ユーザーはグループに存在している限り削除することはできません。

DeleteConflict.User.AccessKey

- ・ HTTPステータスコード: 409
- ・ エラーメッセージ: ユーザーはグループに存在している限り、削除することはできません。

DeleteConflict.User.LoginProfile

- ・ HTTPステータスコード: 409
- ・ エラーメッセージ: ユーザーはログインプロファイルを所有している限り削除することはできません。

DeleteConflict.User.MFADevice

- ・ HTTPステータスコード: 409
- ・ エラーメッセージ: ユーザーはMFAデバイスを所有している限り削除することはできません。

DeleteConflict.User.Policy

- ・ HTTPステータスコード: 409
- ・ ラーメッセージ: ユーザーはポリシーにアタッチされている限り削除することはできません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

```
<DeleteUser Response>
  <RequestId>1C488B66 - B819 - 4D14 - 8711 - C4EAAA13AC 01</
  RequestId>
</DeleteUser Response>
```

- ・ JSON形式

```
{
  "RequestId": "1C488B66 - B819 - 4D14 - 8711 - C4EAAA13AC 01"
}
```

3.5 ユーザーの一覧表示 (ListUsers)

インターフェイス説明

すべての RAM ユーザーの一覧を表示します。

リクエストパラメータ

Action

- ・ 名称: Action
- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: システムパラメータ、必須。パラメータ値: 「ListUsers」。

Marker

- ・ 名称: Marker
- ・ データ型: String
- ・ 必須項目: いいえ

- ・ 説明：返された結果がインターセプトされた場合は、「Marker」を使用して、インターセプトされた続きのコンテンツを取得できます。

MaxItems

- ・ 名称：MaxItems
- ・ データ型：Integer
- ・ 必須項目：いいえ
- ・ 値の範囲：[1 - 1000]
- ・ デフォルト値：「100」。
- ・ 説明：戻り値項目の数。返された結果が「MaxItems」の制限に達して、インターセプトされた場合、リターンパラメータ `IsTruncated` の値は `true` になります。

リターンパラメータ

IsTruncated

- ・ データ型：Boolean
- ・ 説明：返された結果がインターセプトされているかどうかを判断します。

Marker

- ・ データ型：String
- ・ 説明：このパラメータは、`IsTruncated` の値が `true` の場合にのみ使用できます。
`true` の場合は `Marker` を使用して、返された結果のインターセプトされたコンテンツの続きを取得する必要があります。

Users

- ・ データ型：User型の配列User
- ・ 説明：ユーザー情報の集合

必要な権限

Action

```
ram : ListUsers
```

Resource

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

操作の例

リクエストの例

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

&< Public request parameters >

戻り値の例

・ XML形式

```
< ListUsersR  esponse >
  < RequestId > 4B450CA1 - 36E8 - 4AA2 - 8461 - 86B42BF4CC 4E </
  RequestId >
  < Users >
    < User >
      < UserId > 1227489245 380721 </ UserId >
      < UserName > zhangqiang </ UserName >
      < DisplayNam e > zhangqiang </ DisplayNam e >
      < MobilePhon e > 86 - 1860000888 8 </ MobilePhon e >
      < Email > zhangqiang @ example . com </ Email >
      < Comments > This is a cloud computing engineer .</
Comments >
      < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </ CreateDate >
      < UpdateDate > 2015 - 01 - 23T12 : 33 : 18Z </ UpdateDate >
    </ User >
    < User >
      < UserId > 1406498224 724456 </ UserId >
      < UserName > lili </ UserName >
      < DisplayNam e > lili </ DisplayNam e >
      < MobilePhon e > 86 - 1860000999 9 </ MobilePhon e >
      < Email > lili @ example . com </ Email >
      < Comments > Permission administra tor </ Comments >
      < CreateDate > 2015 - 02 - 18T17 : 22 : 08Z </ CreateDate >
      < UpdateDate > 2015 - 02 - 18T17 : 22 : 08Z </ UpdateDate >
    </ User >
  </ Users >
  < IsTruncate d > true </ IsTruncate d >
  < Marker > EXAMPLE </ Marker >
</ ListUsersR  esponse >
```

・ JSON形式

```
{
  " RequestId " : " 4B450CA1 - 36E8 - 4AA2 - 8461 - 86B42BF4CC 4E
",
  " IsTruncate d ": true ,
  " Marker " : " EXAMPLE ",
  " Users " : {
    " User " : [
      {
        " UserId " : " 1227489245 380721 ",
        " UserName " : " zhangqiang ",
        " DisplayNam e ": " zhangqiang ",
        " MobilePhon e ": " 86 - 1860000888 8 ",
        " Email ": " zhangqiang @ example . com ",
        " Comments ": " This is a cloud computing engineer
".
        " CreateDate " : " 2015 - 01 - 23T12 : 33 : 18Z ",
        " UpdateDate " : " 2015 - 01 - 23T12 : 33 : 18Z "
      },
      {
        " UserId " : " 1406498224 724456 ",
        " UserName " : " lili ",
        " DisplayNam e ": " lili ",
        " MobilePhon e ": " 86 - 1860000999 9 ",
        " Email ": " lili @ example . com ",
```

```
    " Comments " : " Permission  administra  tor "
    " CreateDate " : " 2015 - 02 - 18T17 : 22 : 08Z ",
    " UpdateDate " : " 2015 - 02 - 18T17 : 22 : 08Z "
  }
}
}
```

3.6 ログインプロファイルの作成 (CreateLoginProfile)

インターフェイス説明

RAMユーザーのWEBコンソールログインモードを有効にします。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス、必須。パラメータ値：「CreateLoginProfile」。

UserName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ユーザーの名前。

Password

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：パスワード。パスワード強度の要件を満たす必要があります。パスワード強度設定のインターフェイスを無効にするには、[パスワードポリシーの取得 \(GetPasswordPolicy\)](#) を参照してください。

PasswordResetRequired

- ・ データ型：Boolean
- ・ 必須項目：いいえ
- ・ デフォルト値：「False」。
- ・ 説明：ログインの際にパスワードの変更を要求するかどうかを指定します。

MFABindRequired

- ・ データ型：Boolean

- ・ 必須項目：いいえ
- ・ デフォルト値：「False」。
- ・ 説明：次回ログイン時にMFAデバイスのバインドを要求するかどうかを指定します。

リターンパラメータ

LoginProfile

- ・ データ型：[ログインプロファイル \(LoginProfile\)](#)
- ・ 説明：ログインの構成情報。

必要な権限

Action

```
ram : CreateLoginProfile
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

InvalidParameter.Password.TooWeak

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「Password」はパスワードポリシーの要件を満たしていません。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

EntityAlreadyExists.User.LoginProfile

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ユーザーログインプロファイルは既に存在しています。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=CreateLoginProfile
&UserName=zhangqiang
&Password=myspassword
&PasswordRequireSetRequired=true
&<PublicRequestParameters>
```

戻り値の例

・XML形式

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

・JSON形式

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

3.7 ログインプロファイルの表示 (GetLoginProfile)

インターフェイス説明

RAMユーザーのログイン構成を表示します。

リクエストパラメータ

Action

- ・データ型: String
- ・必須項目: はい
- ・説明: 操作インターフェイス、必須。パラメータ値: 「GetLoginProfile」。

UserName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ユーザーの名前。

リターンパラメータ

LoginProfile

- ・ データ型：[ログインプロファイル \(LoginProfile\)](#)
- ・ 説明：ログインの構成情報。

必要な権限

Action

```
ram : GetLoginProfile
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

EntityNotExist.User.LoginProfile

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーログインプロファイルは存在しません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=GetLoginProfile  
&UserName=zhangqiang
```

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

戻り値の例

- ・ XML形式

```
< GetLoginProfile >
  < RequestId > </ RequestId >
  < LoginProfile >
    < Username > </ username >
    < PasswordResetRequired > true </ PasswordResetRequired >
    < MfaBindRequired > true </ mfaBindRequired >
    < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </ CreateDate >
  </ LoginProfile >
</ GetLoginProfile >
```

- ・ JSON形式

```
{
  "RequestId": "04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68",
  "LoginProfile": {
    "Username": "glasqiang",
    "PasswordResetRequired": true,
    "MfaBindRequired": true,
    "CreateDate": "2015 - 01 - 23T12 : 33 : 18Z"
  }
}
```

3.8 ログインプロファイルの削除（DeleteLoginProfile）

インターフェイス説明

RAMユーザーのWEBコンソールログインモードを無効にします。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：システムパラメータ、必須。パラメータ値：「DeleteLoginProfile」。

UserName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ユーザーの名前。例：zhangqiang

リターンパラメータ

パブリックパラメータのみが返されます。詳細は、[パブリックパラメータ](#)を参照してください。

必要な権限

Action

```
ram : DeleteLogi nProfile
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

EntityNotExist.User.LoginProfile

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーログインプロファイルは存在しません。

操作示例

操作の例

```
リクエストの例  
& UserName = zhangqiang  
&< Public request parameters >
```

戻り値の例

- ・ XML形式

```
< DeleteLogi nProfileRe sponse >  
  < RequestId > 1C488B66 - B819 - 4D14 - 8711 - C4EAAA13AC 01 </  
  RequestId >  
  </ FIG >
```

- ・ JSON形式

```
{
```

```
" RequestId ": " 1C488B66 - B819 - 4D14 - 8711 - C4EAAA13AC 01
"
```

3.9 ログインプロファイルの更新 (UpdateLoginProfile)

インターフェイス説明

ユーザーのログイン構成を更新します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス、必須。パラメータ値：「UpdateLoginProfile」。

Username

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ユーザーの名前。

Password

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：パスワード。パスワード強度の要件を満たす必要があります。

PasswordResetRequired

- ・ データ型：Boolean
- ・ 必須項目：いいえ
- ・ 説明：ログインの際にパスワードの変更を要求するかどうかを指定します。

MFABindRequired

- ・ データ型：Boolean
- ・ 必須項目：いいえ
- ・ 説明：次回ログイン時にMFAデバイスのバインドを要求するかどうかを指定します。

リターンパラメータ

LoginProfile

- ・ データ型：LoginProfile型 [ログインプロファイル \(LoginProfile\)](#)
- ・ * 説明：ログインの構成情報。

必要な権限

Action

```
ram : UpdateLogi nProfile
```

Resource

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

エラーメッセージ

Invalidparameter.username.invalidchars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

EntityNotExist.User.LoginProfile

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーログインプロファイルは存在しません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=UpdateLogi nProfile
&UserName=zhangqiang
&Password=mypassword
&PasswordRequire d=true
&<Public request parameter>
```

戻り値の例

- ・ XML形式

```
<UpdateLogi nProfile>
  <RequestId>04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68</
RequestId>
```

```
</ UpdateLogi nProfile >
```

- ・ JSON形式

```
{
  " RequestId ": " 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68
}
```

3.10 AccessKeyの作成 (CreateAccessKey)

インターフェイス説明

RAMユーザーのためのアクセスキーを作成します。

リクエストパラメータ

Action

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 操作インターフェイス、必須。パラメータ値: 「CreateAccessKey」。

Username

- ・ データ型: String
- ・ 必須項目: いいえ
- ・ 説明: ユーザーの名前。サブアカウントでこのインターフェイスを呼び出す場合は、自ずとアクセスキーが作成されます。

リターンパラメータ

AccessKey

- ・ データ型: AccessKey型[アクセスキー \(AccessKey\)](#)
- ・ 説明: アクセスキー。

必要な権限

Action

```
ram : CreateAcce ssKey
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

LimitExceeded.User.AccessKey

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ユーザーのアクセスキーの数は制限を超えています。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

```
<CreateAccessKeyResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <AccessKey>
    <AccessKeyId>0wNEpMMLzy7szvai</AccessKeyId>
    <AccessKeySecret>PupkTg8jdm au1cXxYacg E736PJj4cA</AccessKeySecret>
    <Status>Active</Status>
    <CreateDate>2015-01-23T12:33:18Z</CreateDate>
  </AccessKey>
</CreateAccessKeyResponse>
```

- ・ JSON形式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "AccessKey": {
    "AccessKeyId": "0wNEpMMLzy7szvai",
    "AccessKeySecret": "PupkTg8jdm au1cXxYacg E736PJj4cA",
    "Status": "active",
    "CreateDate": "2015-01-23T12:33:18Z"
  }
}
```

```
}
```

3.11 アクセスキーの更新 (UpdateAccessKey)

インターフェイス説明

RAM ユーザーのアクセスキーのステータスを変更します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス、必須。パラメータ値：「UpdateAccessKey」。

UserAccessKeyId

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：更新したいアクセスキーのID。

Status

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：アクセスキーのステータス。「Active」と「Inactive」の2つのオプションで構成されます。

UserName

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：ユーザーの名前。

リターンパラメータ

パブリックパラメータのみが返されます。詳細は、[パブリックパラメータ](#)を参照してください。

必要な権限

Action

```
ram : UpdateAccessKey
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

EntityNotExist.User.AccessKey

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーアクセスキーは存在しません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=UpdateAccessKey
&UserName=zhangqiang
&UserAccessKeyId=0wNEpMMLzy7szvai
&Status=Inactive
&<Publicrequestparameters>
```

戻り値の例

- ・ XML形式

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

```
</ UpdateAccessKeyResponse >
```

- ・ JSON形式

```
" RequestId ": " 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 "
```

3.12 アクセスキーの削除 (DeleteAccessKey)

インターフェイス説明

RAMユーザーのアクセスキーを削除します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス、必須。 値：「DeleteAccessKey」。

UserAccessKeyId

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：削除したいアクセスキーを指定します。

UserName

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：ユーザーの名前。

リターンパラメータ

パブリックパラメータのみが返されます。 詳細は、次を参照してください。 [パブリックパラメータ](#)

必要な権限

Action

```
ram : DeleteAccessKey
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

EntityNotExist.User.AccessKey

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーアクセスキーは存在しません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=DeleteAccessKey
&UserName=zhangqiang
&UserAccessKeyId=0wNEpMMLzy7szvai
&<Publicrequestparameters>
```

戻り値の例

- ・ XML形式

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

```
</ DeleteAccessKeyResponse >
```

- ・ JSON形式

```
" RequestId ": " 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 "
```

3.13 アクセスキーの一覧表示 (ListAccessKeys)

インターフェイス説明

特定のユーザーのアクセスキーの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 操作インターフェイス、必須。パラメータ値: 「ListAccessKeys」。

UserName

- ・ データ型: String
- ・ 必須項目: いいえ
- ・ 説明: ユーザーの名前。サブアカウントからアクセスしている場合は、このパラメーターを設定していなければ、自分のアクセスキーが一覧表示されます。

リターンパラメータ

AccessKeys

- ・ データ型: [アクセスキー \(AccessKey\)](#) の配列
- ・ 説明: ユーザー情報の集合。

必要な権限

Action

```
ram : ListAccess Keys
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=ListAccessKeys
&<Publicrequestparameters>
```

戻り値の例

- ・ XML形式

```
<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>WnIWUruvfaDT37vQ</AccessKeyId>
      <Status>Inactive</Status>
      <CreateDate>2015-03-24T21:12:21Z</CreateDate>
    </AccessKey>
  </AccessKeys>
</ListAccessKeysResponse>
```

- ・ JSON形式

```
{
  "RequestId": "4B450CA1-36E8-4AA2-8461-86B42BF4CC4E",
  "AccessKeys": {
    "Accesskey": [
      {
        "AccessKeyId": "0wNEpMMLzy7szvai",
        "Status": "Active",
        "CreateDate": "2015-01-23T12:33:18Z"
      },
      {
        "AccessKeyId": "WnIWUruvfaDT37vQ",
        "Status": "Inactive",
        "CreateDate": "2015-03-24T21:12:21Z"
      }
    ]
  }
}
```

3.14 仮想MFAデバイスの作成 (CreateVirtualMFADevice)

インターフェイス説明

仮想MFAデバイスを作成します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス、必須。パラメータ値：「CreateVirtualMFADevice」。

VirtualMFADeviceName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：デバイスの名称。使用可能な文字数は最大64文字です。
- ・ 制限：

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

リターンパラメータ

VirtualMFADevice

- ・ データ型：[仮想MFAデバイス \(VirtualMFADevice\)](#)
- ・ 説明：仮想MFAデバイス。

必要な権限

Action

```
ram : CreateVirtualMFADevice
```

Resource

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

エラーメッセージ

InvalidParameter.VirtualMFADeviceName.InvalidChars

- ・ HTTPステータスコード：400

- ・ エラーメッセージ：パラメータ - 「VirtualMFADeviceName」に無効な文字が含まれています。

InvalidParameter.VirtualMFADeviceName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「VirtualMFADeviceName」は長さの制限を超えています。

EntityAlreadyExists.VirtualMFADevice

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：仮想MFAデバイスは既に存在しています。

LimitExceeded.VirtualMFADevice

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：仮想MFAデバイスの数は制限を超えています。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=CreateVirtualMFADevice
&VirtualMFADeviceName=device001
&<Publicrequestparameters>
```

戻り値の例

- ・ XML形式

```
<CreateVirtualMFADeviceResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</RequestId>
  <VirtualMFADevice>
    <SerialNumber>acs:ram::1234567890123:mfa/device001</VirtualMFADeviceSerialNumber>
    <Base32StringSeed>DSF98HAD982KJA9SDFNA S9D8FU839B8ADHBGSD7A8</Base32StringSeed>
    <QRCodePNG>YXNkZmFzZDlmW5hc2Q5OGZoODd4bXJmcThhaGU5aSBmYXNkZiBzYWVmIGFGIDRxd2VjIG EgdHEzdCAg</QRCodePNG>
  </VirtualMFADevice>
</CreateVirtualMFADeviceResponse>
```

- ・ JSON形式

```
{
  "RequestId": "04F0F334-1335-436C-A1D7-6C044FE73368",
  "VirtualMFADevice": {
    "SerialNumber": "acs:ram::1234567890123:mfa/device001",
    "Base32StringSeed": "DSF98HAD982KJA9SDFNA S9D8FU839B8ADHBGSD7A8",
    "QRCodePNG": "YXNkZmFzZDlmW5hc2Q5OGZoODd4bXJmcThhaGU5aSBmYXNkZiBzYWVmIGFGIDRxd2VjIG EgdHEzdCAg"
```

```

    " Base32StringSeed ": " DSF98HAD98 2KJA9SDFNA
S9D8FU839B 8ADHBGSD7A 8 ",
    " QRCodePNG ": " YXNkZmFzZD lmeW5hc2Q5 OGZoODd4bX
JmcThhaGU5 aSBmYXNkZi BzYWVmIGFG IDRxd2VjIG EgdHEzdCAg "
  }

```

3.15 仮想MFAデバイスの一覧表示 (ListVirtualMFADevices)

インターフェイス説明

仮想MFAデバイスの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 操作インターフェイス、必須。パラメータ値: 「ListVirtualMFADevices」。

リターンパラメータ

VirtualMFADevices

- ・ データ型: [仮想MFAデバイス \(VirtualMFADevice\)](#) の配列
- ・ 説明: 仮想MFAデバイスのリスト。

必要な権限

Action

```
ram : ListVirtualMFADevices
```

Resource

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

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=ListVirtualMFADevices
&<Publicrequestparameters>
```

戻り値の例

- ・ XML形式

```

<ListVirtualMFADevicesResponse>
  <RequestId>04F0F334-1335-436C-A1D7-6C044FE73368</
RequestId>
  <VirtualMFADevices>

```



```

    < VirtualMFA Device >
      < SerialNumb er > acs : ram :: 1234567890 123 : mfa /
device001 </ SerialNumb er >
    </ VirtualMFA Device >
    < VirtualMFA Device >
      < SerialNumb er > acs : ram :: 1234567890 123 : mfa /
device002 </ MFASerialN umber >
      < ActivateDa te > 2015 - 02 - 18T17 : 22 : 08Z <
Activate >
      < User >
        < UserId > 1227489245 380721 </ UserId >
        < UserName > zhangqiang </ UserName >
        < DisplayNam e > zhangqiang </ DisplayNam e >
      </ User >
    </ VirtualMFA Device >
  < VirtualMFA Devices >
</ ListVirtua lMFADevice sResponse >

```

・ JSON形式

```

{
  "RequestId": "04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68",
  "VirtualMFA Devices": {
    "VirtualMFA Device": [
      {
        "SerialNumb er": "acs : ram :: 1234567890 123 : mfa / device001",
        "SerialNumb er": "acs : ram :: 1234567890 123 : mfa / device002",
        "ActivateDa te": "2015 - 02 - 18T17 : 22 : 08Z",
        "User": {
          "UserId": "1227489245 380721",
          "UserName": "zhangqiang",
          "DisplayNam e": "zhangqiang"
        }
      }
    ]
  }
}

```

3.16 仮想MFAデバイスの削除 (DeleteVirtualMFADevice)

インターフェイス説明

仮想MFAデバイスを削除します。

リクエストパラメータ

Action

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 操作インターフェイス、必須。パラメータ値: 「DeleteVirtualMFADevice」。

SerialNumber

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：MFAデバイスのシリアル番号。

リターンパラメータ

パブリックパラメータのみが返されます。詳細は、次を参照してください。[パブリックパラメータ](#)

必要な権限

Action

```
ram : DeleteVirtualMFADevice
```

Resource

```
${ SerialNumber }
```

エラーメッセージ

EntityNotExist.VirtualMFADevice

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：仮想MFAデバイスは存在しません。

DeleteConflict.VirtualMFADevice.User

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：仮想MFAデバイスはユーザーにバインドしている限り、削除することはありません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=DeleteVirtualMFADevice
&SerialNumber=acs:ram::1234567890:123:mfa/device002
&<Publicrequestparameters>
```

戻り値の例

- ・ XML形式

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

```
</ DeleteVirtualMFADeviceResponse >
```

- ・ JSON形式

```
" RequestId ": " 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 "
```

3.17 MFAデバイスのバインド (Bindmfadvice)

インターフェイス説明

MFAデバイスをバインドします。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス。必須。パラメータ値：「BindMFADevice」。

SerialNumber

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：MFAデバイスのシリアル番号。

UserName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ユーザーの名前。

AuthenticationCode1

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：第1動的パスワードを認証します。

AuthenticationCode2

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：第2動的パスワードを認証します。

リターンパラメータ

パブリックパラメータのみが返されます。詳細は、[パブリックパラメータ](#)を参照してください。

必要な権限

Action

```
ram : BindMFADevice
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

InvalidParameter.AuthenticationCode.Format

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「AuthenticationCode」は6桁の数字でなければなりません。

CheckAuthenticationCodeFail

- ・ HTTPステータスコード：403
- ・ エラーメッセージ：認証コードは正しくありません。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

EntityNotExist.VirtualMFADevice

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：仮想MFAデバイスが存在しません。

EntityAlreadyExists.User.MFADevice

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ユーザーにはMFAデバイスをバインド済みです。

EntityAlreadyExists.VirtualMFADevice.User

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：仮想MFAデバイスはユーザーにバインド済みです。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=BindMFADevice
&SerialNumber=acs:ram::1234567890:123:mfa/device002
&Username=zhangqiang
&<Publicrequestparameters>
```

戻り値の例

- ・ XML形式

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

- ・ JSON形式

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

3.18 MFAデバイスのバインド解除 (UnbindMFADevice)

インターフェイス説明

多要素認証デバイスをバインド解除します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必修項目：はい
- ・ 説明：操作インターフェイス、必須。パラメータ値：「UnbindMFADevice」。

UserName

- ・ データ型：String
- ・ 必修項目：はい

- ・ 説明：ユーザーの名前。

リターンパラメータ

MFADevice

- ・ データ型：[MFAデバイス \(MFADevice\)](#)
- ・ 説明：MFAデバイスの情報。

必要な権限

Action

```
ram : UnbindMFAD  evice
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

EntityNotExist.User.MFADevice

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーにはMFAデバイスがバインドされていません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

```
< UnbindMFAD eviceRespo nse >
  < RequestId > 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 </
  RequestId >
  < MFADevice >
    < SerialNumb er > acs : ram :: 1234567890 123 : mfa /
    device002 </ SerialNumb er >
  </ MFADevice >
</ UnbindMFAD eviceRespo nse >
```

- ・ JSON形式

```
" RequestId ": " 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68
",
" MFADevice ": {
  " SerialNumb er ":" acs : ram :: 1234567890 123 : mfa /
  device002 "
}
```

3.19 ユーザーMFA情報を取得（GetUserMFAInfo）

インターフェイス説明

特定のユーザーにバインドしているMFAデバイス情報を取得します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス。必須。パラメータ値：「GetUserMFAInfo」。

UserName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ユーザーの名前。

リターンパラメータ

MFADevice

- ・ データ型：[MFAデバイス（MFADevice）](#)
- ・ 説明：MFAデバイスの情報。

必要な権限

Action

```
ram : GetUserMFA Info
```

Resource

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

エラーメッセージ

EntityNotExist.User

- ・ HTTPステータスコード： 404
- ・ エラーメッセージ：ユーザーは存在しません。

EntityNotExist.User.MFADevice

- ・ HTTPステータスコード： 404
- ・ エラーメッセージ：ユーザーにはMFAデバイスがバインドしていません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

```
< GetUserMFA InfoResponse >
  < RequestId > 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 </
RequestId >
  < MFADevice >
    < SerialNumber > Virtual - MFA - device002 </ SerialNumb
er >
  </ MFADevice >
</ GetUserMFA InfoResponse >
```

- ・ JSON形式

```
{
  "RequestId": "04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68",
  "MFADevice": {
    "SerialNumber": "Virtual - MFA - device002"
  }
}
```



```
}
```

3.20 パスワードの変更 (ChangePassword)

インターフェイス説明

RAMユーザーのパスワードを変更します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス。 必須。 値：ChangePassword。

OldPassword

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：過去のパスワード。

NewPassword

- ・ データ型：String
- ・ 必須項目：はい
- ・ 強度ポリシー要件を満たしたパスワードを設定します。 パスワードの設定インターフェイスの詳細は、[パスワードポリシーの設定 \(SetPasswordPolicy\)](#) を参照してください。

リターンパラメータ

パブリックパラメータのみが返されます。 詳細は、次を参照してください。 [パブリックパラメータ](#)

必要な権限

Action

```
ram : ChangePass word
```

Resource

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

エラーメッセージ

NotSupport.Account

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：このメソッドを、サブユーザーにのみ呼び出されることができます。

InvalidParameter.OldPassword.Incorrect

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：パラメータ - 「OldPassword」は正しくありません。

InvalidParameter.NewPassword.TooWeak

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：パラメータ - 「NewPassword」はパスワードポリシーに準拠していません。

InvalidParameter.NewPassword.ReusePrevention

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：パラメータ - 「NewPassword」は再利用防止 パスワードポリシーに準拠していません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

```
<ChangePassword>
  <RequestId>04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68</
  RequestId>
</ChangePassword>
```

- ・ JSON形式

```
"RequestId": "04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68"
```

4 グループ管理インターフェイス

4.1 グループの作成（CreatGroup）

インターフェイス説明

ユーザーグループを作成します。

リクエストパラメータ

Action

- ・ Type : String
- ・ 必須項目 : はい
- ・ 説明 : 操作の名前。 値 :

```
CreateGroup
```

GroupName

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : グループの名前。 使用できる文字数は最大 64 文字です。
- ・ 形式 :

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

Comments

- ・ データ型 : String
- ・ 必須項目 : いいえ
- ・ 説明 : 補足情報。 使用できる文字数は最大128文字です。

リターンパラメータ

Group

- ・ データ型 : [グループ \(Group\)](#)
- ・ 説明 : グループ情報

必要な権限

Action

```
ram : CreateGroup
```

Resource

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

エラーメッセージ

InvalidParameter.GroupName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」に無効な文字が含まれています。

InvalidParameter.GroupName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」は長さの制限を超えています。

InvalidParameter.Comments.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「Comments」は長さの制限を超えています。

EntityAlreadyExists.Group

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：グループは既に存在します。

LimitExceeded.Group

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：グループ数は制限を超えています。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=CreateGroup
&GroupName=Dev-Team
&Comments=Development team
&<Public request parameters>
```

戻り値の例

- ・ XML形式

```
<CreateGroupResponse>
```

```
< RequestId > D3F0679E - 9757 - 95DB - AF2D - 04D5188C69 C5 </
RequestId >
  < Group >
    < GroupName > Dev - Team </ GroupName >
    < Comments > Developmen t team </ Comments >
    < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </ CreateDate >
  </ Group >
</ CreateGrou pResponse >
```

・ JSON形式

```
" RequestId ": " D3F0679E - 9757 - 95DB - AF2D - 04D5188C69 C5
",
" Group ": {
  " GroupName ": " Dev - Team ",
  " Comments ": " Developmen t team ",
  " CreateDate ": " 2015 - 01 - 23T12 : 33 : 18Z "
}
```

4.2 グループ情報の取得 (GetGroup)

インターフェイス説明

ユーザーグループ情報を取得します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作の名称。 値：

```
GetGroup
```

GroupName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：グループの名称。 例：dev。
- ・ 形式：

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

リターンパラメータ

Group

- ・ データ型：[グループ \(Group\)](#)
- ・ 説明：グループの情報。

必要な権限

Action

```
ram : GetGroup
```

Resource

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

エラーメッセージ

InvalidParameter.GroupName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GropuName」に無効な文字が含まれています。

InvalidParameter.GroupName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」は長さの制限を超えています。

EntityNotExist.Group

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：グループは存在しません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=GetGroup
&GroupName=Dev-Team
&<Public request parameters>
```

戻り値の例

- ・ XML形式

```
< GetGroupResponse >
  < RequestId > D4065824 - E422 - 3ED6 - 68B1 - 1AF7D5C780 4C </
RequestId >
  < Group >
    < GroupName > Dev - Team </ GroupName >
    < Comments > Developmen t team </ Comments >
    < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </ CreateDate >
    < UpdateDate > 2015 - 02 - 11T03 : 15 : 21Z </ UpdateDate >
  </ Group >
</ GetGroupResponse >
```

- ・ JSON形式

```
" RequestId ": " D4065824 - E422 - 3ED6 - 68B1 - 1AF7D5C780 4C
",
" Group ": {
  " GroupName ": " Dev - Team ",
  " Comments ": " Developmen t team ",
  " CreateDate ": " 2015 - 01 - 23T12 : 33 : 18Z ",
  " UpdateDate ": " 2015 - 02 - 11T03 : 15 : 21Z "
}
```

4.3 グループ情報の更新 (UpdateGroup)

インターフェイス説明

ユーザーグループ情報を更新します。

リクエストパラメータ

Action

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : 操作の名称。 値 :

```
UpdateGroup
```

GroupName

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : グループの名称。
- ・ 形式 :

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

NewGroupName

- ・ データ型 : String
- ・ 必須項目 : いいえ
- ・ 説明 : 新規作成のグループ名。
- ・ 形式 :

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

NewComments

- ・ データ型 : String
- ・ 必須項目 : いいえ

- ・ 説明：新しい備考情報。 使用できる文字数は最大 128 文字です。

リターンパラメータ

Group

- ・ データ型：[グループ \(Group\)](#)
- ・ 説明：グループの情報。

必要な権限

Action

```
ram : UpdateGroup
```

Resource

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

エラーメッセージ

InvalidParameter.GroupName.InvalidChars

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：パラメータ - 「GroupName」 に無効な文字が含まれています。

InvalidParameter.GroupName.Length

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：パラメータ - 「GroupName」 は長さの制限を超えています。

InvalidParameter.NewGroupName.InvalidChars

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：パラメータ - 「NewGroupName」 に無効な文字が含まれています。

InvalidParameter.NewGroupName.Length

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：パラメータ - 「NewGroupName」 は長さの制限を超えています。

InvalidParameter.NewComments.Length

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：パラメータ - 「NewComments」 は長さの制限を超えています。

EntityNotExist.Group

- ・ HTTPステータスコード： 404

- ・ エラーメッセージ：グループは存在しません。

EntityAlreadyExists.Group

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：グループは既に存在しています。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=UpdateGroup
&GroupName=Dev-Team
&NewGroupName=NewDev-Team
&<PublicRequestParameters>
```

戻り値の例

- ・ XML形式

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

- ・ JSON形式

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

4.4 すべてのユーザーグループを一覧表示 (ListGroups)

インターフェイス説明

すべてのユーザーグループの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型：String

- ・ 必須項目：はい
- ・ 説明：操作の名称。 値：

ListGroups

Marker

- ・ 名称：Marker
- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：返された結果がインターセプトされた場合は、「Marker」を使用して、インターセプトされた戻り値の続きのコンテンツを取得できます。

MaxItems

- ・ 名称：MaxItems
- ・ データ型：Integer
- ・ 必須項目：いいえ
- ・ 値の範囲：「1~1000」
- ・ デフォルト値：100
- ・ 説明：戻り値の項目の数。返された結果が「MaxItems」の制限に達し、インターセプトされた場合、リターンパラメータ `IsTruncated` の値は `true` になります。

リターンパラメータ

Groups

- ・ データ型：グループ (Group) の配列
- ・ 説明：グループ情報の集合

IsTruncated

- ・ データ型：Boolean
- ・ 説明：戻り値がインターセプトされているかどうかを判断します。

Marker

- ・ データ型：String
- ・ 説明：このパラメータは、パラメータ `IsTruncated` の値が `true` の場合にのみ使用できます。`true` の場合、このインターフェイスを起動し、`Marker` を使用して、インターセプトされた部分の続きのコンテンツを取得する必要があります。

必要な権限

Action

```
ram : ListGroups
```

Resource

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

操作の例

リクエストの例

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

戻り値の例

・ XML形式

```
< ListGroups Response >
  < RequestId > 065527AA - 2F2E - AD7C - 7484 - F2626CFE49 34 </
RequestId >
  < IsTruncate d > true </ IsTruncate d >
  < Marker > EXAMPLE </ Marker >
  < Groups >
    < Group >
      < GroupName > Dev - Team </ GroupName >
      < Comments > Developmen t team </ Comments >
      < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </ CreateDate >
      < UpdateDate > 2015 - 01 - 23T12 : 33 : 18Z </ UpdateDate >
    </ Group >
    < Group >
      < GroupName > QA - Team </ GroupName >
      < Comments > Test team </ Comments >
      < CreateDate > 2015 - 02 - 18T17 : 22 : 08Z </ CreateDate >
      < UpdateDate > 2015 - 02 - 18T17 : 22 : 08Z </ UpdateDate >
    </ Group >
  </ Groups >
</ ListGroups Response >
```

・ JSON形式

```
{
  "RequestId": "065527AA - 2F2E - AD7C - 7484 - F2626CFE49 34",
  "IsTruncate d": true,
  "Marker": "EXAMPLE",
  "Groups": {
    "Group": [
      {
        "GroupName": "Dev - Team",
        "Comments": "Developmen t team",
        "CreateDate": "2015 - 01 - 23T12 : 33 : 18Z",
        "UpdateDate": "2015 - 01 - 23T12 : 33 : 18Z"
      },
      {
        "GroupName": "QA - Team",
```

```
" Comments " : " Test team ",
" CreateDate " : " 2015 - 02 - 18T17 : 22 : 08Z ",
" UpdateDate " : " 2015 - 02 - 18T17 : 22 : 08Z "
```

4.5 グループの削除 (DeleteGroup)

インターフェイス説明

特定のグループを削除します。

- ・ グループを削除する前に、グループに権限付与ポリシーがアタッチされていないことを確認する必要があります。
- ・ グループを削除する前に、グループにユーザーが含まれていないことを確認する必要があります。

リクエストパラメータ

Action

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : 操作の名称。 値 :

```
DeleteGroup
```

GroupName

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : グループの名称。 例 : dev。
- ・ 形式 :

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

リターンパラメータ

パブリックパラメーターのみが返されます。 詳細は、次を参照してください。 [パブリックパラメータ](#)

必要な権限

Action

```
ram : DeleteGroup
```

Resource

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

エラーメッセージ

InvalidParameter.GroupName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」に無効な文字が含まれています。

InvalidParameter.GroupName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」は長さの制限を超えています。

EntityNotExist.Group

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：グループは存在しません。

DeleteConflict.Group.User

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：グループにユーザーメンバーがいる限り、グループを削除することはできません。

DeleteConflict.Group.Policy

- ・ HTTPステータスコード：409
- ・ エンティティにポリシーがアタッチされている限り、グループを削除することはできません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=DeleteGroup  
&GroupName=Dev-Team  
&<Public request parameters>
```

戻り値の例

- ・ XML形式

```
< DeleteGroupResponse >
  < RequestId > FCF40AB5 - 881C - A0F9 - 334C - B0AD423AA6 9D </
  RequestId >
</ DeleteGroupResponse >
```

- ・ JSON形式

```
" RequestId ": " FCF40AB5 - 881C - A0F9 - 334C - B0AD423AA6 9D
"
```

4.6 ユーザーをユーザーグループに追加 (AddUserToGroup)

インターフェイス説明

ユーザーを特定のユーザーグループに追加します。

リクエストパラメータ

Action

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 操作の名称。 値:

```
AddUserToGroup
```

UserName

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: ユーザーの名前。

GroupName

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: グループの名称。

リターンパラメータ

パブリックパラメーターのみが返されます。 詳細は、次を参照してください。 [パブリックパラメータ](#)

必要な権限

Action

```
ram : AddUserToG roup
```

Resource

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

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

InvalidParameter.GroupName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GropuName」に無効な文字が含まれています。

InvalidParameter.GroupName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

EntityNotExist.Group

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：グループは存在しません。

EntityAlreadyExists.User.Group

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ユーザーはグループに参加済みです。

LimitExceeded.User.Group

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ターゲットユーザーが参加したグループの数が制限を超えています。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=AddUserToGroup
&UserName=zhangqiang
&GroupName=Dev-Team
&<PublicRequestParameters>
```

戻り値の例

- ・ XML形式

```
<AddUserToGroupResponse>
  <RequestId>1B968853-B423-63A6-FE1F-45E81BC2AD-61</
  RequestId>
</AddUserToGroupResponse>
```

- ・ JSON形式

```
"RequestId": "1B968853-B423-63A6-FE1F-45E81BC2AD-61"
```

4.7 ユーザーグループからユーザーを削除 (RemoveUserFromGroup)

インターフェイス説明

ユーザーグループからユーザーを削除します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作の名称。 値：

```
RemoveUserFromGroup
```

UserName

- ・ データ型：String
- ・ 必須項目：はい

- ・ 説明：ユーザーの名前。

GroupName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：グループの名称。

リターンパラメータ

パブリックパラメータのみが返されます。詳細は、次を参照してください。[パブリックパラメータ](#)

必要な権限

Action

```
ram : RemoveUser FromGroup
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

InvalidParameter.GroupName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GropuName」に無効な文字が含まれています。

InvalidParameter.GroupName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

EntityNotExist.Group

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：グループは存在しません。

EntityNotExist.User.Group

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーはグループに参加していません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=RemoveUserFromGroup
&UserName=zhangqiang
&GroupName=Dev-Team
&<Publicrequestparameters>
```

戻り値の例

- ・ XML形式

```
<RemoveUserFromGroupResponse>
  <RequestId>A07EF215-B9B3-8CB2-2899-3F9575C6E3-20</
  RequestId>
</RemoveUserFromGroupResponse>
```

- ・ JSON形式

```
"RequestId": "A07EF215-B9B3-8CB2-2899-3F9575C6E3-20"
```

4.8 ユーザーの所属グループ一覧を表示 (ListGroupForUser)

インターフェイス説明

特定のユーザーが参加しているグループの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい

- ・ 説明：操作の名称。 値：

```
ListGroups ForUser
```

UserName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ユーザーの名前。

リターンパラメータ

Groups

- ・ データ型：グループ (Group) の配列
- ・ 説明：グループ情報の集合。

必要な権限

Action

```
ram : ListGroups ForUser
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=ListGroupsForUser  
&UserName=zhangqiang
```

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

戻り値の例

・ XML形式

```
< ListGroups ForUserResponse >
  < RequestId > 0945BF6F - 54A4 - C384 - C89C - C500DCEE36 7C </
  RequestId >
  < Groups >
    < Group >
      < GroupName > Dev - Team </ GroupName >
      < Comments > Development team </ Comments >
      < JoinDate > 2015 - 01 - 23T12 : 33 : 18Z </ JoinDate >
    </ Group >
    < Group >
      < GroupName > QA - Team </ GroupName >
      < Comments > Test team </ Comments >
      < JoinDate > 2015 - 02 - 18T17 : 22 : 08Z </ JoinDate >
    </ Group >
  </ Groups >
</ ListGroups ForUserResponse >
```

・ JSON形式

```
{
  "RequestId": "0945BF6F - 54A4 - C384 - C89C - C500DCEE36 7C",
  "Groups": {
    "Group": [
      {
        "GroupName": "Dev - Team",
        "Comments": "Development team",
        "JoinDate": "2015 - 01 - 23T12 : 33 : 18Z"
      },
      {
        "GroupName": "QA - Team",
        "Comments": "Test team",
        "JoinDate": "2015 - 02 - 18T17 : 22 : 08Z"
      }
    ]
  }
}
```

4.9 グループ内ユーザーの一覧表示 (ListUsersForGroup)

インターフェイス説明

特定のユーザーグループ内のユーザーの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい

- ・ 説明：操作の名称。 値：

ListUsersF orGroup

GroupName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：グループの名称。 例：Dev-Team

Marker

- ・ 名称：Marker
- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：返された結果がインターセプトされた場合は、「Marker」を使用して、インターセプトされた戻り値の続きのコンテンツを取得できます。

MaxItems

- ・ 名称：MaxItems
- ・ データ型：Integer
- ・ 必須項目：いいえ
- ・ 値の範囲：「1~1000」
- ・ デフォルト値：100
- ・ 説明：戻り値の数。 返された結果が「MaxItems」の制限に達し、インターセプトされた場合、リターンパラメータ IsTruncated の値は true になります。

リターンパラメータ

Users

- ・ データ型：Userの配列
- ・ 説明：ユーザー情報の集合

IsTruncated

- ・ データ型：Boolean
- ・ 説明：戻り値がインターセプトされているかどうかを判断します。

Marker

- ・ データ型：String

- ・ 説明：このパラメータは、パラメータ `IsTruncated` の値が `true` の場合にのみ使用できます。 `true` の場合、このインターフェイスを起動し、`Marker` を使用して、インターセプトされた部分の続きのコンテンツを取得する必要があります。

必要な権限

Action

```
ram : ListUsersF orGroup
```

Resource

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

エラーメッセージ

EntityNotExist.Group

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：グループは存在しません。

InvalidParameter.GroupName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」に無効な文字が含まれています。

InvalidParameter.GroupName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」は長さの制限を超えています。

EntityNotExist.Group

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：グループは存在しません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=ListUsersF orGroup
&GroupName=dev
&<Public request parameters>
```

戻り値の例

- ・ XML形式

```
<ListUsersF orGroupResponse>
  <RequestId> 5756784B - 79C4 - E82E - 24C2 - FC3E171E5A B3 </
  RequestId >
```

```
< Users >
  < User >
    < UserId > 1227489245 380721 </ UserId >
    < UserName > zhangqiang </ UserName >
    < DisplayName e > zhangqiang </ DisplayName e >
    < JoinDate > 2015 - 01 - 23T12 : 33 : 18Z </ JoinDate >
  </ User >
  < User >
    < UserId > 1406498224 724456 </ UserId >
    < UserName > lili </ UserName >
    < DisplayName e > lili </ DisplayName e >
    < JoinDate > 2015 - 02 - 18T17 : 22 : 08Z </ JoinDate >
  </ User >
</ Users >
</ ListUsersForGroupResponse >
```

・ JSON形式

```
{
  "RequestId" : "4B450CA1 - 36E8 - 4AA2 - 8461 - 86B42BF4CC 4E",
  "Users" : {
    "User" : [
      {
        "UserId" : "1227489245 380721",
        "UserName" : "zhangqiang",
        "DisplayName" : "zhangqiang",
        "JoinDate" : "2015 - 01 - 23T12 : 33 : 18Z"
      },
      {
        "UserId" : "1406498224 724456",
        "UserName" : "lili",
        "DisplayName" : "lili",
        "JoinDate" : "2015 - 02 - 18T17 : 22 : 08Z"
      }
    ]
  }
}
```

5 ロール管理インターフェイス

5.1 ロールの作成 (CreateRole)

インターフェイス説明

ロールを作成します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス、必須。パラメータ値：「CreateRole」。

RoleName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ロールの名称。使用できる文字数は最大 64 文字です。
- ・ 形式：

```
^[ a - zA - Z0 - 9 \.
```

Description

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：ロールの説明。使用できる文字数は最大 1,024 文字です。

AssumeRolePolicyDocument

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ロールを引き受けた一人以上の主体（プリンシパル）を指定するポリシーテキストです。主体（プリンシパル）は、RAMユーザー、またはAlibaba Cloudサービスである必要があります。

- ・ 例：

- 次のポリシーは、このロールを引き受けた主体（プリンシパル）がクラウドアカウント(AccountID=123456789012345678)に属する権限付与されたRAMユーザーであることを示しています。

```
" Statement ": [  
  " Action ": " sts : AssumeRole ",  
  " Effect ": " Allow ",  
  " Principal ": {  
    " RAM ": [  
      " acs : ram :: 1234567890 12345678 : root "  
    ]  
  }  
  
" Version ": " 1 "
```

- 次のポリシーは、このロールを引き受けた主体（プリンシパル）が現在のテナントアカウントに属するECSインスタンスであることを示しています。（ユーザーはECSインスタンスの作成時にRAMロールの使用を指定することができます。そうすることによって、インスタンスの起動後、このRAMロールのSTSトークンを取得できます。）

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

リターンパラメータ

Role

- ・ データ型：[ロール \(Role\)](#)
- ・ 説明：ロールの情報。

必要な権限。**Action**

```
ram : CreateRole
```

Resource

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

エラーメッセージ**InvalidParameter.RoleName.InvalidChars**

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：パラメータ - 「RoleName」に無効な文字が含まれています。

InvalidParameter.RoleName.Length

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：パラメータ - 「RoleName」は長さの制限を超えています。

InvalidParameter.AssumeRolePolicyDocument.Length

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：パラメータ - 「AssumeRolePolicyDocument」は長さの制限を超えています。

EntityAlreadyExists.Role

- ・ HTTPステータスコード： 409
- ・ エラーメッセージ：ロールは既に存在しています。

MalformedPolicyDocument

- ・ HTTPステータスコード： 400
- ・ エラーメッセージ：{エラーの詳細}

LimitExceeded.Role

- ・ HTTPステータスコード： 409
- ・ エラーメッセージ：ロールの数は制限を超えています。

操作の例**リクエストの例**

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

```
& AssumeRole PolicyDocu ment ={ " Statement ": [ { " Action ": " sts
: AssumeRole ", " Effect ": " Allow ", " Principal ": { " RAM ": "
acs : ram :: 1234567890 12345678 : root " } } ], " Version ": " 1
" }
& Descriptio n = ECS administra tor role
&< Public request parameters >
```

戻り値の例

・ XML形式

```
< CreateRole Response >
  < RequestId > 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 </
RequestId >
  < Role >
    < RoleId > 9012345678 90123 </ RoleId >
    < RoleName > ECSAdmin </ RoleName >
    < Arn > acs : ram :: 1234567890 123456 : role / ECSAdmin </
Arn >
    < Descriptio n > ECS Administra tor role </
Descriptio n >
    < AssumeRole PolicyDocu ment >{ " Statement ": [ { "
Action ": " sts : AssumeRole ", " Effect ": " Allow ", " Principal
": { " RAM ": " acs : ram :: 1234567890 12345678 : root " } } ], "
Version ": " 1 " }</ AssumeRole PolicyDocu ment >
    < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </ CreateDate >
  </ Role >
</ CreateRole Response >
```

・ JSON形式

```
" RequestId ": " 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68
",
" Role ": {
  " RoleId ": " 9012345678 90123 ",
  " RoleName ": " ECSAdmin ",
  " Arn ": " acs : ram :: 1234567890 123456 : role / ECSAdmin
",
  " Descriptio n ": " ECS Administra tor role ",
  " AssumeRole PolicyDocu ment ": "{ \" Statement \": [ { \"
Action \": \" sts : AssumeRole \", \" Effect \": \" Allow \", \"
Principal \": { \" RAM \": \" acs : ram :: 1234567890 12345678 :
root \" } } ], \" Version \": \" 1 \" }",
  " CreateDate ": " 2015 - 01 - 23T12 : 33 : 18Z "
```

5.2 ロールの取得 (GetRole)

インターフェイス説明

ロールを取得します。

リクエストパラメータ

Action

- ・ データ型 : String

- ・ 必須項目：はい
- ・ 説明：操作インターフェイス、必須。パラメータ値：「GetRole」。

RoleName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ロールの名称。使用できる文字数は最大 64 文字です。
- ・ 形式：

```
^[ a - zA - Z0 - 9 \.
```

リターンパラメータ

Role

- ・ データ型：ロール (Role)
- ・ 説明：ロールの情報。

必要な権限

Action

```
ram : GetRole
```

Resource

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

エラーメッセージ

InvalidParameter.RoleName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「RoleName」に無効な文字が含まれています。

InvalidParameter.RoleName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「RoleName」は長さの制限を超えています。

EntityNotExist.Role

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ロールは存在しません。

操作の例

リクエストの例

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

戻り値の例

・ XML形式

```
< GetRoleResponse >
  < RequestId > 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 </
  RequestId >
  < Role >
    < RoleId > 9012345678 90123 </ RoleId >
    < RoleName > ECSAdmin </ RoleName >
    < Arn > acs : ram :: 1234567890 123456 : role / ECSAdmin </
    Arn >
    < Description > ECS Administrator role </
    Description >
    < AssumeRolePolicyDocument >{ "Statement": [ { "
    Action": "sts : AssumeRole", "Effect": "Allow", "Principal
    ": { "RAM": "acs : ram :: 1234567890 12345678 : root" } } ], "
    Version": "1" }</ AssumeRole PolicyDocument >
    < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </ CreateDate >
    < UpdateDate > 2015 - 01 - 23T12 : 33 : 18Z </ UpdateDate >
  </ Role >
</ GetRoleResponse >
```

・ JSON形式

```
{
  "RequestId": "04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68",
  "Role": {
    "RoleId": "9012345678 90123",
    "RoleName": "ECSAdmin",
    "Arn": "acs : ram :: 1234567890 123456 : role / ECSAdmin",
    "Description": "ECS Administrator role",
    "AssumeRolePolicyDocument": "{ \"Statement\": [ { \"
    Action\": \"sts : AssumeRole\", \"Effect\": \"Allow\", \"
    Principal\": { \"RAM\": \"acs : ram :: 1234567890 12345678 :
    root\" } } ], \"Version\": \"1\" }",
    "CreateDate": "2015 - 01 - 23T12 : 33 : 18Z",
    "UpdateDate": "2015 - 01 - 23T12 : 33 : 18Z"
  }
}
```

5.3 ロールの更新 (UpdateRole)

インターフェイス説明

ロールを更新します。

リクエストパラメータ

Action

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : 操作インターフェイス、必須。パラメータ値 : 「UpdateRole」。

RoleName

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : ロールの名称 使用できる文字数は最大 64 文字です。
- ・ 形式

```
^[ a - zA - Z0 - 9 \.
```

NewAssumeRolePolicyDocument

- ・ データ型 : String
- ・ 必須項目 : いいえ
- ・ 説明 : ロールの ID を指定します。

リターンパラメータ

Role

- ・ データ型 : [ロール \(Role\)](#)
- ・ 説明 : ロールの情報。

必要な権限

Action

```
ram : UpdateRole
```

Resource

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

エラーメッセージ

InvalidParameter.RoleName.InvalidChars

- ・ HTTPステータスコード : 400
- ・ エラーメッセージ : パラメータ - 「RoleName」に無効な文字が含まれています。

InvalidParameter.RoleName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「RoleName」は長さの制限を超えています。

InvalidParameter.NewAssumeRolePolicyDocument.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「NewAssumeRolePolicyDocument」は長さの制限を超えています。

EntityNotExist.Role

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ロールは存在しません。

MalformedPolicyDocument

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：{エラーの詳細}

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=UpdateRole
&RoleName=ECSAdmin
&NewAssumeRolePolicyDocument=xxxx // TODO
&<Public request parameters>
```

戻り値の例

- ・ XML形式

```
< UpdateRole Response >
  < RequestId > 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 </
  RequestId >
  < Role >
    < RoleId > 9012345678 90123 </ RoleId >
    < RoleName > ECSAdmin </ RoleName >
    < Arn > acs : ram :: 1234567890 123456 : role / ECSAdmin </
    Arn >
    < Description > ECS Administrator role </
    Description >
    < AssumeRole PolicyDocument >{ " Statement ": [ { "
    Action ": " sts : AssumeRole ", " Effect ": " Allow ", " Principal
    ": { " RAM ": " acs : ram :: 1234567890 12345678 : root " } } ], "
    Version ": " 1 " }</ AssumeRole PolicyDocument >
    < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </ CreateDate >
    < UpdateDate > 2015 - 01 - 23T12 : 33 : 18Z </ UpdateDate >
  </ Role >
</ UpdateRole Response >
```

- ・ JSON形式

```

    " RequestId ": " 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68
  ",
  " Role ": {
    " RoleId ": " 9012345678 90123 ",
    " RoleName ": " ECSAdmin ",
    " Arn ": " acs : ram :: 1234567890 123456 : role / ECSAdmin
  ",
    " Descriptio n ": " ECS  Administra tor  role ",
    " AssumeRole PolicyDocu ment ": "{ \" Statement \": [ { \"
Action \": \" sts : AssumeRole \", \" Effect \": \" Allow \", \"
Principal \": { \" RAM \": \" acs : ram :: 1234567890 12345678 :
root \" } } ], \" Version \": \" 1 \" }",
    " CreateDate ": " 2015 - 01 - 23T12 : 33 : 18Z ",
    " UpdateDate ": " 2015 - 01 - 23T12 : 33 : 18Z "
  }
}

```

5.4 ロールの一覧表示 (ListRoles)

インターフェイス説明

ロールの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : 必須。パラメータ値 : 「ListRoles」。

リターンパラメータ

Roles

- ・ データ型 : [ロール \(Role\)](#) の配列
- ・ 説明 : ロール名のリスト。

必要な権限

Action

```
ram : ListRoles
```

Resource

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

操作の例

リクエストの例

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



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

戻り値の例

・ XML形式

```
< ListRolesResponse >
  < RequestId > 7B8A4E7D - 6CFF - 471D - 84DF - 195A7A241E CB </
RequestId >
  < Roles >
    < Role >
      < RoleId > 9012345678 90123 </ RoleId >
      < RoleName > ECSAdmin </ RoleName >
      < Arn > acs : ram :: 1234567890 123456 : role /
ECSAdmin </ Arn >
      < Description > ECS Administrator role </
Description >
      < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </
CreateDate >
      < UpdateDate > 2015 - 01 - 23T12 : 33 : 18Z </
UpdateDate >
    </ Role >
    < Role >
      < RoleId > 9012345678 90456 </ RoleId >
      < RoleName > OSSReadOnlyAccess </ RoleName >
      < Arn > acs : ram :: 1234567890 123456 : role /
OSSReadOnlyAccess </ Arn >
      < Description > OSS read - only access role </
Description >
      < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </
CreateDate >
      < UpdateDate > 2015 - 01 - 23T12 : 33 : 18Z </
UpdateDate >
    </ Role >
  </ Roles >
</ ListRolesResponse >
```

・ JSON形式

```
{
  "RequestId": "7B8A4E7D - 6CFF - 471D - 84DF - 195A7A241E CB",
  "Roles": {
    "Role": [
      {
        "RoleId": "9012345678 90123",
        "RoleName": "ECSAdmin",
        "Arn": "acs : ram :: 1234567890 123456 : role / ECSAdmin",
        "Description": "ECS Administrator role",
        "CreateDate": "2015 - 01 - 23T12 : 33 : 18Z",
        "UpdateDate": "2015 - 01 - 23T12 : 33 : 18Z"
      },
      {
        "RoleId": "9012345678 90456",
        "RoleName": "OSSReadOnlyAccess",
        "Arn": "acs : ram :: 1234567890 123456 : role / OSSReadOnlyAccess",
        "Description": "OSS administrator read - only permission",
        "CreateDate": "2015 - 01 - 23T12 : 33 : 18Z",
        "UpdateDate": "2015 - 01 - 23T12 : 33 : 18Z"
      }
    ]
  }
}
```

5.5 ロールの削除 (DeleteRole)

インターフェイス説明

特定のロールを削除します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「DeleteRole」。

RoleName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ロールの名称。

リターンパラメータ

パブリックパラメータのみが返されます 詳細は、次を参照してください。 [パブリックパラメータ](#)

必要な権限

Action

```
ram : DeleteRole
```

Resource

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

エラーメッセージ

InvalidParameter.RoleName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「RoleName」に無効な文字が含まれています。

InvalidParameter.RoleName.Length

- ・ HTTPステータスコード：400

- ・ エラーメッセージ：パラメータ - 「RoleName」は長さの制限を超えています。

EntityNotExist.Role

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ロールは存在しません。

DeleteConflict.Role.Policy

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ロールにポリシーがアタッチされている限り、削除することはできません。

リクエストの例

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

戻り値の例

- ・ XML形式

```
<DeleteRole Response>
  <RequestId>898FAB24 - 7509 - 43EE - A287 - 086FE4C443 94</
  RequestId>
</DeleteRole Response>
```

- ・ JSON形式

```
"RequestId": "898FAB24 - 7509 - 43EE - A287 - 086FE4C443 94"
```

6 権限付与ポリシー管理インターフェイス

6.1 権限付与ポリシーの作成 (CreatePolicy)

インターフェイス説明

権限付与ポリシーを作成します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「CreatePolicy」。

PolicyName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名称。使用できる文字数は最大 128 文字です。
- ・ 形式：

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

Description

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：権限付与ポリシーの説明。使用できる文字数は最大 1024 文字です。

PolicyDocument

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの内容。使用できる文字数は最大 2048 文字です。

リターンパラメータ

Policy

- ・ データ型：[権限付与ポリシー \(Policy\)](#)
- ・ 説明：権限付与ポリシーの内容。

必要な権限

Action

```
ram : CreatePolicy
```

Resource

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

エラーメッセージ

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

InvalidParameter.PolicyDocument.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyDocument」は長さの制限を超えています。

InvalidParameter.Description.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「Description」は長さの制限を超えています。

MalformedPolicyDocument

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：{エラーの詳細}

EntityAlreadyExists.Policy

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：グループは既に存在しています。

LimitExceeded.Policy

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ポリシーの数は制限を超えています。

リクエストの例

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

戻り値の例

- ・ XML形式

```
<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形式

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

6.2 権限付与ポリシー情報の取得 (GetPolicy)

インターフェイス説明

特定の権限付与ポリシーの情報を取得します。

リクエストパラメータ

Action

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 必須。パラメータ値: 「GetPolicy」。

PolicyType

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーのタイプ。 値：「System」または「Custom」。

PolicyName

- ・ 名称：PolicyName
- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名前。

リターンパラメータ

Policy

- ・ データ型：権限付与ポリシー (Policy)
- ・ 説明：権限付与ポリシーの基本情報。

必要な権限

Action

```
ram : GetPolicy
```

Resource

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

エラーメッセージ

InvalidParameter.PolicyType

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyType」は正しくありません。

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

EntityNotExist.Policy

- ・ HTTPステータスコード：404

- ・ エラーメッセージ：ポリシーは存在しません。

リクエストの例

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

戻り値の例

- ・ XML形式

```
<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形式

```
{
  "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
  }
}
```

6.3 権限付与ポリシーの削除 (DeletePolicy)

インターフェイス説明

特定の権限付与ポリシーを削除します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「DeletePolicy」。

PolicyName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名前。

リターンパラメータ

パブリックパラメータのみが返されます。詳細は、次を参照してください。[パブリックパラメータ](#)

必要な権限

Action

```
ram : DeletePolicy
```

Resource

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

エラーメッセージ

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

DeleteConflict.Policy.User

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ポリシーはユーザーにアタッチしている限り削除することはできません。

DeleteConflict.Policy.Group

- ・ HTTPステータスコード：409
- ・ ポリシーはグループにアタッチしている限り削除することはできません。

DeleteConflict.Policy.Version

- ・ HTTPステータスコード：409
- ・ ポリシーにはデフォルトバージョン以外のバージョンは存在してはなりません。

リクエストの例

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

戻り値の例

- ・ XML形式

```
<DeletePolicyResponse>
  <RequestId>898FAB24-7509-43EE-A287-086FE4C443-94</
  RequestId>
</DeletePolicyResponse>
```

- ・ JSON形式

```
"RequestId": "898FAB24-7509-43EE-A287-086FE4C443-94"
```

6.4 権限付与ポリシーの一覧表示 (ListPolicies)

インターフェイス説明

権限付与ポリシーの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「ListPolicies」。

PolicyType

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：権限付与ポリシーのタイプ。値：「System」または「Custom」。このパラメーターが設定されていない場合は、すべての権限付与ポリシーの一覧が表示されます。

Marker

- ・ 名称：Marker
- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：戻り値の結果がインターセプトされた場合は、「Marker」を使用して、インターセプトされたコンテンツの続きを取得できます。

MaxItems

- ・ 名称：MaxItems
- ・ データ型：Integer
- ・ 必須項目：いいえ
- ・ 値の範囲：「1~1000」
- ・ デフォルト値：100
- ・ 説明: 戻り値の項目の数。返される結果が「MaxItems」の上限に達してインターセプトされた場合、リターンパラメータ `IsTruncated` の値は `true` になります。

リターンパラメータ

IsTruncated

- ・ データ型：Boolean
- ・ 説明：戻り値がインターセプトされたかどうかを示します。

Marker

- ・ データ型：String
- ・ 説明：このパラメータは、パラメーター `IsTruncated` の値が `true` の場合にのみ使用できます。値が `true` の場合は、このインターフェイスを起動し、`Marker` を使用して、返された結果のインターセプトされたコンテンツの続きを取得する必要があります。

Policies

- ・ データ型：権限付与ポリシー (Policy) の配列
- ・ 説明：権限付与ポリシー名のリスト。

必要な権限

Action

```
ram : ListPolicies
```

Resource

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

リクエストの例

```
https://ram.aliyuncs.com/?Action=ListPolicies
&<PublicRequestParameters>
```

戻り値の例

・ XML形式

```
<ListPoliciesResponse>
  <RequestId> 7B8A4E7D - 6CFF - 471D - 84DF - 195A7A241E CB </
  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形式

```
{
  "RequestId": " 7B8A4E7D - 6CFF - 471D - 84DF - 195A7A241E CB",
  "IsTruncated": true,
  "Marker": " EXAMPLE ",
  "Policies": {
```

```

    " Policy ": [
      " PolicyName ": " OSS - Administra  tor ",
      " PolicyType ": " Custom ",
      " Descriptio  n ": " OSS  administra  tor
permission ",
      " DefaultVer  sion ": " v1 ",
      " CreateDate ": " 2015 - 01 - 23T12 : 33 : 18Z ",
      " UpdateDate ": " 2015 - 01 - 23T12 : 33 : 18Z ",
      " Attachment  Count ": 3

      " PolicyName ": " ReadOnlyAc  cess ",
      " PolicyType ": " System ",
      " Descriptio  n ": " Read - only  permission ",
      " DefaultVer  sion ": " v1 ",
      " CreateDate ": " 2015 - 02 - 11T18 : 39 : 12Z ",
      " UpdateDate ": " 2015 - 02 - 19T09 : 43 : 16Z ",
      " Attachment  Count ": 1
    ]
  }
}

```

6.5 権限付与ポリシーのバージョンを作成 (CreatePolicyVersion)

インターフェイス説明

権限付与ポリシーのバージョンを作成します。

リクエストパラメータ

Action

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 必須。パラメータ値: 「CreatePolicyVersion」。

PolicyName

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 権限付与ポリシーの名前。

PolicyDocument

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 権限付与ポリシーの内容。 使用できる文字数は最大2048文字です。

SetAsDefault

- ・ データ型：String
- ・ 必須項目：いいえ
- ・ 説明：デフォルトの権限ポリシーとして設定するかどうかを指定できます。デフォルト値は「false」です。

リターンパラメータ

PolicyVersion

- ・ データ型：[ポリシーバージョン \(PolicyVersion\)](#)
- ・ Description:新規作成した権限付与ポリシーのバージョン情報。

必要な権限

Action

```
ram : CreatePolicyVersion
```

Resource

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

エラーメッセージ

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

LimitExceeded.Policy.Version

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ポリシーバージョンの数は制限を超えています。

リクエストの例

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

```
& Policydocument = {" statement ": [{" action ": [" Oss : * " ],"
  effect ":" allow "," resource ": [" ACS : OSS : * " ]}], "
  version ":" 1 "}
&< Public request parameters >
```

戻り値の例

- ・ XML形式

```
< CreatePolicyVersionResponse >
  < RequestId > 9B34724D - 54B0 - 4A51 - B34D - 4512372FE1 BE </
RequestId >
  < PolicyVersion >
    < VersionId > v3 </ VersionId >
    < IsDefaultVersion > false </ IsDefaultVersion >
    < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </ CreateDate >
  </ Policy >
</ CreatePolicyVersionResponse >
```

- ・ JSON形式

```
{
  " RequestId ": " 9B34724D - 54B0 - 4A51 - B34D - 4512372FE1 BE ",
  " PolicyVersion ": {
    " VersionId ": " v3 ",
    " IsDefaultVersion ": false ,
    " CreateDate ": " 2015 - 01 - 23T12 : 33 : 18Z "
  }
}
```

6.6 権限付与ポリシーのバージョンを取得 (GetPolicyVersion)

インターフェイス説明

権限付与ポリシーのバージョンを取得します。

リクエストパラメータ

Action

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : 必須。パラメータ値 : 「GetPolicyVersion」。

PolicyType

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : 権限付与ポリシーのタイプ。値 : 「System」または「Custom」。

PolicyName

- ・ データ型 : String

- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名前。

VersionId

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ターゲットバージョンの ID。

リターンパラメータ

PolicyVersion

- ・ 名前： [ポリシーバージョン \(PolicyVersion\)](#)
- ・ データ型：String
- ・ 説明：権限付与ポリシーのバージョン情報。

必要な権限

Action

```
ram : GetPolicyVersion
```

Resource

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

エラーメッセージ

InvalidParameter.PolicyType

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyType」は正しくありません。

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

InvalidParameter.VersionId.Format

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「VersionId」は正しくありません。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

EntityNotExist.Policy.Version

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーバージョンは存在しません。

操作の例**リクエストの例**

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

戻り値の例

- ・ XML形式

```
<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形式

```
{
  "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\"}"
  }
}
```

```
}
```

6.7 権限付与ポリシーのバージョンを削除 (DeletePolicyVersion)

インターフェイス説明

特定の権限付与ポリシーバージョンを削除します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「DeletePolicyVersion」。

PolicyName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーのタイプ。

VersionId

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ターゲットバージョンの ID。

リターンパラメータ

パブリックパラメータのみが戻ります。詳細は、次を参照してください。[パブリックパラメータ](#)

必要な権限

Action

```
ram : DeletePolicyVersion
```

Resource

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

エラーメッセージ

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400

- ・ エラーメッセージ：パラメータ - 「GroupName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」は長さの制限を超えています。

InvalidParameter.VersionId.Format

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「VersionId」は正しくありません。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

EntityNotExist.Policy.Version

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーバージョンは存在しません。

DeleteConflict.Policy.Version.Default

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：デフォルトのポリシーバージョンは直接削除することはできません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

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

- ・ JSON形式

```
" RequestId ": " 9B34724D - 54B0 - 4A51 - B34D - 4512372FE1 BE
"
```

6.8 権限付与ポリシーのバージョンの一覧表示 (ListPolicyVersions)

インターフェイス説明

権限付与ポリシーのバージョンの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「ListPolicyVersions」。

PolicyType

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーのタイプ。値：「System」または「Custom」。

PolicyName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名前。

リターンパラメータ

PolicyVersions

- ・ データ型：[ポリシーバージョン \(PolicyVersion\)](#) の配列
- ・ 説明：権限付与ポリシー名のリスト。

必要な権限

Action

```
ram : ListPolicy Versions
```

Resource

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

エラーメッセージ

InvalidParameter.PolicyType

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyType」は正しくありません。

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

```
< ListPolicy VersionsRe sponse >  
  < RequestId > 7B8A4E7D - 6CFF - 471D - 84DF - 195A7A241E CB </  
  RequestId >  
    < PolicyVers ions >  
      < PolicyVers ion >  
        < VersionId > v3 </ VersionId >  
        < IsDefaultV ersion > false </ IsDefaultV ersion >  
        < CreateDate > 2015 - 01 - 23T12 : 33 : 18Z </  
      CreateDate >
```

```

    </ PolicyVers ion >
    < PolicyVers ion >
      < VersionId > v5 </ VersionId >
      < IsDefaultV ersion > true </ IsDefaultV ersion >
      < CreateDate > 2015 - 02 - 26T01 : 25 : 52Z </
CreateDate >
    </ PolicyVers ion >
  </ PolicyVers ions >
</ ListPolicy VersionsRe sponse >

```

・ JSON形式

```

{
  " RequestId ": " 7B8A4E7D - 6CFF - 471D - 84DF - 195A7A241E CB",
  " Policyvers ions ":{
    " PolicyVers ion ": [
      {
        " VersionId ": " v3 ",
        " IsDefaultV ersion ": false ,
        " CreateDate ": " 2015 - 01 - 23T12 : 33 : 18Z "
      },
      {
        " VersionId ": " v5 ",
        " IsDefaultV ersion ": true ,
        " CreateDate ": " 2015 - 02 - 26T01 : 25 : 52Z "
      }
    ]
  }
}

```

6.9 デフォルト権限付与ポリシーバージョンの設定 (SetDefaultPolicyVersion)

インターフェイス説明

デフォルトの権限付与ポリシーバージョンを設定します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。 パラメータ値：「SetDefaultPolicyVersion」。

PolicyName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名前。

VersionId

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：新規作成したデフォルトバージョンのID。

リターンパラメータ

パブリックパラメータのみが返されます。詳細について、次を参照してください。[パブリックパラメータ](#)

必要な権限

Action

```
ram : SetDefault PolicyVersion
```

Resource

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

エラーメッセージ

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

InvalidParameter.VersionId.Format

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「VersionId」は正しくありません。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

EntityNotExist.Policy.Version

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーバージョンは存在しません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

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

- ・ JSON形式

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

6.10 ユーザーに権限付与ポリシーをアタッチ (AttachPolicyToUser)

インターフェイス説明

特定のユーザーに権限付与ポリシーをアタッチします。

リクエストパラメータ

Action

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 必須。パラメータ値: 「AttachPolicyToUser」。

PolicyType

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 権限付与ポリシーのタイプ。値: 「System」または「Custom」。

PolicyName

- ・ データ型: String
- ・ 必須項目: はい

- ・ 説明：権限付与ポリシーの名前。

UserName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ユーザーの名前。例：zhangqiang

リターンパラメータ

パブリックパラメータのみが返されます。詳細について、次を参照してください。[パブリックパラメータ](#)

必要な権限

Action

```
ram : AttachPolicyToUser
```

Resource

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

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

InvalidParameter.PolicyType

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyType」は正しくありません。

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

EntityAlreadyExists.User.Policy

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ユーザーに既にこのポリシーがアタッチされています。

LimitExceeded.User.Policy

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ユーザーのポリシーの数は制限を超えています。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

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

- ・ JSON形式

```
" RequestId ": " 697852FB - 50D7 - 44D9 - 9774 - 530C31EAC5 72 "
```

6.11 ユーザーから権限付与ポリシーをデタッチ (DetachPolicyFromUser)

インターフェイス説明

特定の権限付与ポリシーをユーザーからデタッチします。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「DetachPolicyFromUser」。

PolicyType

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーのタイプ。値：「System」または「Custom」。

PolicyName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名前。

UserName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ユーザーの名前。例：zhangqiang

リターンパラメータ

パブリックパラメータのみが返されます。詳細について、次を参照してください。[パブリックパラメータ](#)

必要な権限

Action

```
ram : DetachPolicyFromUser
```

Resource

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

エラーメッセージ

InvalidParameter.PolicyType

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyType」は正しくありません。

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ユーザーは存在しません。

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

EntityNotExist.User.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ: 指定のユーザーポリシーは存在しません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

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

- ・ JSON形式

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

6.12 グループに権限付与ポリシーをアタッチ (AttachPolicyToGroup)

インターフェイス説明

特定のユーザーグループに権限付与ポリシーをアタッチします。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「AttachPolicyToGroup」。

PolicyType

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーのタイプ。値：「System」または「Custom」。

PolicyName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名前。

GroupName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：グループの名前。 例：dev。

リターンパラメータ

パブリックパラメータのみが返されます。 詳細は、次を参照してください。 [パブリックパラメータ](#)

必要な権限

Action

```
ram : AttachPolicyToGroup
```

Resource

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

エラーメッセージ

InvalidParameter.GroupName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」に無効な文字が含まれています。

InvalidParameter.GroupName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」は長さの制限を超えています。

InvalidParameter.PolicyType

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyType」は正しくありません。

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400

- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

EntityNotExist.Group

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：グループは存在しません。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

EntityAlreadyExists.Group.Policy

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：グループに既にこのポリシーがアタッチされています。

LimitExceeded.Group.Policy

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：グループのポリシーの数は制限を超えています。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

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

- ・ JSON形式

```
" RequestId ": " 697852FB - 50D7 - 44D9 - 9774 - 530C31EAC5 72 "
```

6.13 グループから権限付与ポリシーをデタッチ (DetachPolicyFromGroup)

インターフェイス説明

指定された権限付与ポリシーをグループからデタッチします。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「DetachPolicyFromGroup」。

PolicyType

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーのタイプ。値：「System」または「Custom」。

PolicyName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名前。

GroupName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：グループの名前。例：dev。

リターンパラメータ

パブリックパラメータのみが返されます。詳細は、次を参照してください。[パブリックパラメータ](#)

必要な権限

Action

```
ram : DetachPolicyFromGroup
```

Resource

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

エラーメッセージ

InvalidParameter.PolicyType

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyType」は正しくありません。

InvalidParameter.GroupName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」に無効な文字が含まれています。

InvalidParameter.GroupName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」は長さの制限を超えています。

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

EntityNotExist.Group

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：グループは存在しません。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

EntityNotExist.Group.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：グループにアタッチした指定のポリシーは存在しません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

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

- ・ JSON形式

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

6.14 ロールに権限付与ポリシーをアタッチ (AttachPolicyToRole)

インターフェイス説明

特定のロールに権限付与ポリシーをアタッチします。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値は「AttachPolicyToRole」です。

PolicyType

- ・ データ型：String
- ・ 必須項目：Yes
- ・ 説明：権限付与ポリシー 値：「System」または「Custion」です。

PolicyName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名前。

RoleName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ロールの名前。例：dev。

リターンパラメータ

パブリックパラメータのみが返されます。詳細は、次を参照してください。[パブリックパラメータ](#)

必要な権限

Actonm

```
ram : AttachPolicyToRole
```

Resource

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

エラーメッセージ

InvalidParameter.RoleName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「RoleName」に無効な文字が含まれています。

InvalidParameter.RoleName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「RoleName」は長さの制限を超えています。

InvalidParameter.PolicyType

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyType」は正しくありません。

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400

- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

EntityNotExist.Role

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ロールは存在しません。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

EntityAlreadyExists.Role.Policy

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ロールには既にこのポリシーがアタッチされています。

LimitExceeded.Role.Policy

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：ロールのポリシーの数は制限を超えています。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

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

- ・ JSON形式

```
" RequestId ": " 697852FB - 50D7 - 44D9 - 9774 - 530C31EAC5 72 "
```

6.15 ロールからの権限付与ポリシーのデタッチ (DetachPolicyFromRole)

インターフェイス説明

指定された権限付与ポリシーをロールからデタッチします。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「DetachPolicyFromRole」。

PolicyType

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーのタイプ。値：「System」または「Custom」。

PolicyName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名前。

RoleName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ロールの名前。例：dev。

リターンパラメータ

パブリックパラメータのみが戻ります。詳細は、次を参照してください。[パブリックパラメータ](#)

必要な権限

Action

```
ram : DetachPolicyFromRole
```

Resource

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

エラーメッセージ

InvalidParameter.RoleName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「RoleName」に無効な文字が含まれています。

InvalidParameter.RoleName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「RoleName」は長さの制限を超えています。

InvalidParameter.PolicyType

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyType」は正しくありません。

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PolicyName」は長さの制限を超えています。

EntityNotExist.Role

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ロールは存在しません。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

EntityNotExist.Role.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ: ロールにアタッチした指定のポリシーは存在しません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

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

- ・ JSON形式

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

6.16 ユーザーに付与された権限付与ポリシーの一覧表示 (ListPoliciesForUser)

インターフェイス説明

ユーザーに付与された権限付与ポリシーの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「ListPoliciesForUser」。

UserName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ユーザーの名前。例：zhangqiang

リターンパラメータ

Policies

- ・ データがt: **権限付与ポリシー (Policy)** の配列
- ・ 説明: 権限付与ポリシー名のリスト。

エラーメッセージ

InvalidParameter.UserName.InvalidChars

- ・ HTTPステータスコード: 400
- ・ エラーメッセージ: パラメータ - 「UserName」に無効な文字が含まれています。

InvalidParameter.UserName.Length

- ・ HTTPステータスコード: 400
- ・ エラーメッセージ: パラメータ - 「UserName」は長さの制限を超えています。

EntityNotExist.User

- ・ HTTPステータスコード: 404
- ・ エラーメッセージ: ユーザーは存在しません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=ListPoliciesForUser
&<Publicrequestparameters>
```

戻り値の例

- ・ XML形式

```
<ListPoliciesForUserResponse>
  <RequestId>7B8A4E7D-6CFF-471D-84DF-195A7A241E CB</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 >
</ ListPoliciesForUserResponse >

```

・ JSON形式

```

{
  "RequestId": "7B8A4E7D - 6CFF - 471D - 84DF - 195A7A241E CB",
  "Policies": {
    "Policy": [
      {
        "PolicyName": "OSS - Administrator",
        "PolicyType": "Custom",
        "Description": "OSS administrator",
        "DefaultVersion": "v1",
        "AttachDate": "2015 - 01 - 23T12 : 33 : 18Z",
        "Permission": "permission"
      },
      {
        "PolicyName": "ReadOnlyAccess",
        "PolicyType": "System",
        "Description": "Read-only permission",
        "DefaultVersion": "v1",
        "AttachDate": "2015 - 02 - 18T17 : 22 : 08Z"
      }
    ]
  }
}

```

6.17 権限付与ポリシーを参照するエンティティの一覧表示 (ListEntitiesForPolicy)

インターフェイス説明

権限付与ポリシーを参照するエンティティの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 必須。パラメータ値: 「ListEntitiesForPolicy」。

PolicyType

- ・ データ型: String
- ・ 必須項目: はい
- ・ 説明: 権限付与ポリシーのタイプ。値: 「System」または「Custom」。

PolicyName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：権限付与ポリシーの名前。

リターンパラメータ

Groups

- ・ データ型：グループ (Group) の配列
- ・ 説明：グループ情報の集合

Users

- ・ データ型：Userの配列
- ・ 説明：ユーザー情報の集合

Roles

- ・ データ型：ロール (Role) の配列
- ・ 説明：ロール情報の集合。

必要な権限

Action

```
ram : ListEntitiesForPolicy
```

Resource

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

エラーメッセージ

InvalidParameter.PolicyType

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PlicyType」は正しくありません。

InvalidParameter.PolicyName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PlicyName」に無効な文字が含まれています。

InvalidParameter.PolicyName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「PlicyName」は長さの制限を超えています。

EntityNotExist.Policy

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：ポリシーは存在しません。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

```
<ListEntitiesForPolicyResponse>
  <RequestId> 7B8A4E7D - 6CFF - 471D - 84DF - 195A7A241E CB </
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> 1227489245 380721 </UserId>
      <UserName> zhangqiang </UserName>
      <DisplayName> zhangqiang </DisplayName>
      <AttachDate> 2015 - 01 - 23T12 : 33 : 18Z </
AttachDate>
    </User>
    <User>
      <UserId> 1406498224 724456 </UserId>
      <UserName> lili </UserName>
      <DisplayName> lili </DisplayName>
      <AttachDate> 2015 - 02 - 18T17 : 22 : 08Z </
AttachDate>
    </User>
  </Users>
  <Roles>
    <Role>
      <RoleId> 9012345678 90123 </RoleId>
      <RoleName> ECSAdmin </RoleName>
      <Arn> acs : ram :: 1234567890 123456 : role /
ECSAdmin </Arn>
      <Description> ECS Administrator role </
Description>
      <AttachDate> 2015 - 01 - 23T12 : 33 : 18Z </
AttachDate>
    </Role>
```

```

    < Role >
      < RoleId > 9012345678 90456 </ RoleId >
      < RoleName > OSSReadonlyAccess </ RoleName >
      < Description > OSS read - only access role </
Description >
      < Arn > acs : ram :: 1234567890 123456 : role /
OSSReadonlyAccess </ Arn >
      < AttachDate > 2015 - 01 - 23T12 : 33 : 18Z </
AttachDate >
    </ Role >
  </ Roles >
</ ListEntitiesForPolicyResponse >

```

・ JSON形式

```

" RequestId " : " 7B8A4E7D - 6CFF - 471D - 84DF - 195A7A241E CB
",
" 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 " : " 1227489245 380721 ",
      " Username " : " zhangqiang ",
      " DisplayName " : " zhangqiang ",
      " AttachDate " : " 2015 - 01 - 23T12 : 33 : 18Z "

      " UserId " : " 1406498224 724456 ",
      " Username " : " lili ",
      " DisplayName " : " lili ",
      " AttachDate " : " 2015 - 02 - 18T17 : 22 : 08Z "

    " Roles " : {
      " Role " : [

        " RoleId " : " 1227489245 380721 ",
        " RoleName " : " ECSAdmin ",
        " Arn " : " acs : ram :: 1234567890 123456 : role / ECSAdmin
",
        " Description " : " ECS Administrator role ",
        " AttachDate " : " 2015 - 01 - 23T12 : 33 : 18Z "

        " RoleId " : " 1406498224 724456 ",
        " RoleName " : " OSSReadonlyAccess ",
        " Arn " : " acs : ram :: 1234567890 123456 : role /
OSSReadonlyAccess ",
        " Description " : " OSS administrator read - only
permission ",

```

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

6.18 グループに付与された権限付与ポリシーの一覧表示 (ListPoliciesForGroup)

インターフェイス説明

グループに付与された権限付与ポリシーの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメーター値：「ListPoliciesForGroup」。

GroupName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：グループの名前。例：dev。

リターンパラメータ

ポリシー

- ・ データ型：権限付与ポリシー (Policy) の配列
- ・ 説明: 権限付与ポリシー名のリスト。

エラーメッセージ

InvalidParameter.GroupName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」に無効な文字が含まれています。

InvalidParameter.GroupName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「GroupName」は長さの制限を超えています。

EntityNotExist.Group

- ・ HTTPステータスコード：404
- ・ エラーメッセージ：グループは存在しません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=ListPoliciesForGroup
&<PublicRequestParameters>
```

戻り値の例

・ XML形式

```
<ListPoliciesForGroupResponse>
  <RequestId>7B8A4E7D-6CFF-471D-84DF-195A7A241E-CB</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形式

```
{
  "RequestId": "7B8A4E7D-6CFF-471D-84DF-195A7A241E-CB",
  "Policies": [
    {
      "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"
    }
  ]
}
```

6.19 ロールに付与された権限付与ポリシーの一覧表示 (ListPoliciesForRole)

インターフェイス説明

ロールに付与された権限付与ポリシーの一覧を表示します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：必須。パラメータ値：「ListPoliciesForRole」。

RoleName

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：ロールの名前 例：AdminRole。

リターンパラメータ

ポリシー

- ・ データ型：権限付与ポリシー (Policy) の配列
- ・ 説明：権限付与ポリシー名のリスト。

エラーメッセージ

InvalidParameter.RoleName.InvalidChars

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「RoleName」に無効な文字が含まれています。

InvalidParameter.RoleName.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「RoleName」は長さの制限を超えています。

EntityNotExist.Role

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：ロールは存在しません。

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=ListPoliciesForRole
&<PublicRequestParameters>
```

戻り値の例

・ XML形式

```
<ListPoliciesForRoleResponse>
  <RequestId>7B8A4E7D-6CFF-471D-84DF-195A7A241E-CB</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形式

```
{
  "RequestId": "7B8A4E7D-6CFF-471D-84DF-195A7A241E-CB",
  "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"
      }
    ]
  }
}
```


7 セキュリティ管理 API

7.1 アカウントエイリアスの設定 (SetAccountAlias)

インターフェイス説明

Alibaba Cloud アカウントエイリアスを設定します。

リクエストパラメータ

Action

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : 操作インターフェイス、必須。パラメータ値 : 「SetAccountAlias」。

AccountAlias

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : Alibaba Cloud アカウントエイリアス。3~32文字で構成しています。
- ・ 制限 :

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

リターンパラメータ

パブリックパラメータのみが戻ります。詳細について、次を参照してください。[パブリックパラメータ](#)

必要な権限

Action

```
ram : SetAccount Alias
```

Resource

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

エラーメッセージ

InvalidParameter.AccountAlias.Format

- ・ HTTPステータスコード : 400

- ・ エラーメッセージ：パラメータ形式 - 「AccountAlias」は正しくありません。

InvalidParameter.AccountAlias.Length

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「AccountAlias」の範囲は3~32です。

EntityAlreadyExists.AccountAlias

- ・ HTTPステータスコード：409
- ・ エラーメッセージ：アカウントエイリアスはすでに存在しています。

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

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

- ・ JSON形式

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

7.2 アカウントエイリアスの表示 (GetAccountAlias)

インターフェイス説明

Alibaba Cloudアカウントのエイリアスを表示します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス。必須。パラメータ値：「GetAccountAlias」。

リターンパラメータ

AccountAlias

- ・ データ型 : String
- ・ 説明 : Alibaba Cloud アカウントのエイリアス。

必要な権限

Action

```
ram : GetAccount Alias
```

Resource

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

操作の例

リクエストの例

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

戻り値の例

- ・ XML形式

```
< GetAccount AliasResponse >  
  < RequestId > 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 </  
  RequestId >  
  < AccountAlias > myalias </ AccountAlias >  
</ GetAccount AliasResponse >
```

- ・ JSON形式

```
{  
  "RequestId": "04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68",  
  "AccountAlias": "myalias"  
}
```

7.3 アカウントエイリアスの削除 (ClearAccountAlias)

インターフェイス説明

Alibaba Cloud アカウントエイリアスを削除します。

リクエストパラメータ

Action

- ・ データ型 : String

- ・ 必須項目：はい
- ・ 説明：操作インターフェイス、システム指定パラメータ。パラメータ値：「ClearAccountAlias」。

戻り値

パブリックパラメータのみが戻ります。詳細について、次を参照してください。[パブリックパラメータ](#)

必要な権限

Action

```
ram : ClearAccountAlias
```

Resource

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

操作の例

リクエストの例

```
https://ram.aliyuncs.com/?Action=ClearAccountAlias
&<Publicrequestparameters>
```

戻り値の例

- ・ XML形式

```
<ClearAccountAliasResponse>
  <RequestId>04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68</
  RequestId>
</ClearAccountAliasResponse>
```

- ・ JSON形式

```
"RequestId": "04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68"
```

7.4 パスワードポリシーの設定 (SetPasswordPolicy)

インターフェイス説明

ユーザーのパスワードポリシー (パスワードの強度など) を設定します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス、必須。パラメータ値：「SetPasswordPolicy」。

MinimumPasswordLength

- ・ データ型：Integer
- ・ 必須項目：いいえ
- ・ 説明：パスワードの最小文字数。

RequireLowercaseCharacters

- ・ データ型：Boolean
- ・ 必須項目：いいえ
- ・ 説明：小文字を含める必要があります。

RequireUppercaseCharacters

- ・ データ型：Boolean
- ・ 必須項目：いいえ
- ・ 説明：大文字を含める必要があります。

RequireNumbers

- ・ データ型：Boolean
- ・ 必須項目：いいえ
- ・ 説明：数字を含める必要があります。

RequireSymbols

- ・ データ型：Boolean
- ・ 必須項目：いいえ
- ・ 説明：特殊文字を含める必要があります。

戻り値

PasswordPolicy

- ・ データ型：[パスワードポリシー \(PasswordPolicy\)](#)
- ・ 説明：パスワードポリシー。

必要な権限

Action

```
ram : SetPasswor dPolicy
```

Resource

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

エラーメッセージ

InvalidParameter.MinimumPasswordLength.Range

- ・ HTTPステータスコード：400
- ・ エラーメッセージ：パラメータ - 「MinimumPasswordLength」の範囲は8～32 です。

操作の例

リクエストの例

```
https :// ram . aliyuncs . com /? Action = SetPasswor dPolicy
& MinimumPas swordLengt h = 12
& RequireLow ercaseChar acters = true
& RequireUpp ercaseChar acters = true
& RequireNum bers = true
& RequireSym bols = true
&< Public request parameters >
```

戻り値の例

- ・ XML形式

```
< SetPasswor dPolicyRes ponse >
  < RequestId > 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 </
RequestId >
  < PasswordPo licy >
    < MinimumPas swordLengt h > 12 </ MinimumPas swordLengt
h >
    < RequireLow ercaseChar acters > true </ RequireLow
ercaseChar acters >
    < RequireUpp ercaseChar acters > true </ RequireUpp
ercaseChar acters >
    < RequireNum bers > true </ RequireNum bers >
    < RequireSym bols > true </ RequireSym bols >
  </ PasswordPo licy >
</ SetPasswor dPolicyRes ponse >
```

- ・ JSON形式

```
" RequestId ": " 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68
",
" PasswordPo licy ": {
  " MinimumPas swordLengt h ": 12 ,
  " RequireLow ercaseChar acters ": true ,
```

```

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

```

7.5 パスワードポリシーの取得 (GetPasswordPolicy)

インターフェイス説明

ユーザーパスワードポリシー（パスワードの強度など）を取得します。

リクエストパラメータ

Action

- ・ データ型：String
- ・ 必須項目：はい
- ・ 説明：操作インターフェイス、システム指定のパラメータ。パラメータ値: 「SetSecurityPreference」。

リターンパラメータ

PasswordPolicy

- ・ データ型：[パスワードポリシー \(PasswordPolicy\)](#)
- ・ 説明：パスワードポリシー。

必要な権限

Action

```
ram : GetPasswordPolicy
```

Resource

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

操作の例

リクエストの例

```

https://ram.aliyuncs.com/?Action=GetPasswordPolicy
&AccountAlias=myalias
&<Publicrequestparameters>

```

戻り値の例

- ・ XML形式

```
<GetPasswordPolicyResponse>
```

```
< RequestId > 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 </
RequestId >
  < PasswordPolicy >
    < MinimumPasswordLength > 12 </ MinimumPasswordLength
  >
    < RequireLowercaseCharacters > true </ RequireLower
    caseCharacters >
    < RequireUppercaseCharacters > true </ RequireUpp
    ercaseCharacters >
    < RequireNumbers > true </ RequireNumbers >
    < RequireSymbols > true </ RequireSymbols >
  </ PasswordPolicy >
</ GetPasswordPolicyResponse >
```

・ JSON形式

```
" RequestId ": " 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68
",
" PasswordPolicy ": {
  " MinimumPasswordLength ": 12 ,
  " RequireLowercaseCharacters ": true ,
  " RequireUppercaseCharacters ": true ,
  " RequireNumbers ": true ,
  " RequireSymbols ": true
}
```

7.6 グローバルセキュリティプリファレンスの設定 (SetSecurityPreference)

インターフェイス説明

グローバルセキュリティプリファレンスを設定します

リクエストパラメータ

Action

- ・ データ型 : String
- ・ 必須項目 : はい
- ・ 説明 : 操作インターフェイス。 システム指定のパラメータ。 値 : SetSecurityPreference。

EnableSaveMFATicket

- ・ データ型 : Boolean
- ・ 必須項目 : いいえ
- ・ 説明 : ユーザーにログイン時に自らMFA権限付与チケットの保存を許可するか否かを指定します。 現在のチケット有効期間は7日間です。

AllowUserToChangePassword

- ・ データ型 : Boolean

- ・ 必須項目：false
- ・ デフォルト：true
- ・ 説明：ユーザーに自らパスワードの変更を許可するか否かを指定します。

AllowUserToManageAccessKeys

- ・ データ型：Boolean
- ・ 必須項目：false
- ・ デフォルト：false
- ・ 説明：ユーザーに自らアクセスキーの管理を許可するか否かを指定します。

AllowUserToManageMFADevices

- ・ データ型：Boolean
- ・ 必須項目：false
- ・ デフォルト：true
- ・ 説明：ユーザーに自らMFAデバイスのバインド/バインド解除を許可するか否かを指定します。

リターンパラメータ

SecurityPreference

- ・ データ型：グローバルセキュリティプリファレンスの設定 (SetSecurityPreference)
- ・ 説明：

必要な権限

Action

```
ram : SetSecurityPreference
```

Resource

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

操作の例

操作の例

```
https://ram.aliyuncs.com/?Action=SetSecurityPreference
&EnableSaveMFATicket=true
&AllowUserToChangePassword=true
&AllowUserToManageAccessKeys=false
```

&< Common Request Parameter >

戻り値の例

・ XML形式

```
< SetSecurityPreferenceResponse >
  < RequestId > 04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68 </
  RequestId >
  < SecurityPreference >
    < LoginProfilePreference >
      < EnableSaveMFATicket > true </ EnableSaveMFATicket
    >
      < AllowUserToChangePassword > true </ AllowUserTo
    ChangePassword >
    </ LoginProfilePreference > LoginProfilePreference >
    < AccessKeyPreference >
      < AllowUserToManageAccessKeys > false </ AllowUserTo
    ManageAccessKeys >
    </ AccessKeyPreference >
    < MFAPreference >
      < AllowUserToManageMFADevices > false </ AllowUserTo
    ManageMFADevices >
    </ MFAPreference >
    </ SecurityPreference >
  </ SetSecurityPreferenceResponse >
```

・ JSON形式

```
{
  "RequestId": "04F0F334 - 1335 - 436C - A1D7 - 6C044FE733 68",
  "SecurityPreference": {
    "LoginProfilePreference": {
      "EnableSaveMFATicket": true,
      "AllowUserToChangePassword": true
    },
    "AccessKeyPreference": {
      "AllowUserToManageAccessKeys": false
    },
    "MFAPreference": {
      "AllowUserToManageMFADevices": true
    }
  }
}
```