

Resource Access Manager

API Reference

Issue 01
Date 2022-10-29



Copyright © Huawei Technologies Co., Ltd. 2022. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Before You Start.....	1
1.1 Overview.....	1
1.2 API Calling.....	1
1.3 Endpoints.....	1
1.4 Concepts.....	1
2 API Overview.....	3
3 Calling APIs.....	4
3.1 Making an API Request.....	4
3.2 Authentication.....	7
3.3 Response.....	8
4 API.....	11
4.1 RAM Managed Permissions.....	11
4.1.1 Getting a List of RAM Managed Permissions.....	11
4.1.2 Getting Details About RAM Managed Permissions.....	14
4.2 Resource Shares.....	15
4.2.1 Creating a Resource Share.....	16
4.2.2 Searching for a Resource Share.....	19
4.2.3 Updating a Resource Share.....	23
4.2.4 Deleting a Resource Share.....	25
4.3 Associated Principals and Resources.....	26
4.3.1 Associating Principals and Resources.....	26
4.3.2 Disassociating Principals and Resources.....	29
4.3.3 Searching for Associated Principals and Resources.....	31
4.4 Associated RAM Managed Permissions.....	34
4.4.1 Associating or Replacing RAM Managed Permissions.....	34
4.4.2 Disassociating RAM Managed Permissions.....	36
4.4.3 Getting Associated RAM Managed Permissions.....	37
4.5 Shared Resources.....	40
4.5.1 Searching for Shared Resources.....	40
4.6 Principals.....	44
4.6.1 Searching for Principals.....	44
4.7 Resource Sharing Invitations.....	46

4.7.1 Accepting a Resource Sharing Invitation.....	47
4.7.2 Rejecting a Resource Sharing Invitation.....	49
4.7.3 Searching for a Resource Sharing Invitation.....	51
4.8 Sharing with Organizations.....	54
4.8.1 Checking Whether Sharing with Organizations Is Enabled.....	54
4.8.2 Enabling Sharing with Organizations.....	55
4.8.3 Disabling Sharing with Organizations.....	56
5 Permissions Policies and Supported Actions.....	58
5.1 Introduction.....	58
5.2 Actions.....	59
6 Appendixes.....	62
6.1 Status Codes.....	62
6.2 Error Codes.....	65
6.3 Obtaining Information About Account, IAM User, Group, Project, Region, and Agency.....	71
7 Change History.....	73

1 Before You Start

1.1 Overview

Welcome to use the Resource Access Manager (RAM) service. RAM is a service that helps you securely share your cloud resources across Huawei Cloud accounts. You can use RAM to share the resources that you create in one of your Huawei Cloud accounts with other Huawei Cloud accounts. With RAM, you can also centrally manage shared resources.

This document describes RAM application programming interfaces (APIs), including the API descriptions, parameters, and examples. You can use the APIs to perform RAM-related operations, such as creating, deleting, modifying, and querying. For details about all supported operations, see [API Overview](#).

If you plan to use an API to access RAM, ensure that you are familiar with RAM concepts described in [Service Overview](#).

1.2 API Calling

RAM supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For more details, see [Calling APIs](#).

1.3 Endpoints

An endpoint is the request address for calling an API. Endpoints vary depending on services and regions. For the RAM endpoints, see [Regions and Endpoints](#).

1.4 Concepts

- Account

An account is created upon successful registration with HUAWEI CLOUD. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity and should not be used directly to perform

routine management. For security purposes, create IAM users and grant them permissions for routine management.

- User

An IAM user is created using an account to use cloud services. Each IAM user has their own identity credentials (password and access keys).

An IAM user can view the account ID and user ID on the **My Credentials** page of the console. The account name, username, and password will be required for API authentication.

- Region

Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into Global region and specific regions. The Global region provides common cloud services for all tenants, and a specific region provides services of the same type or provides special services for specific tenants.

For details, see [Region and AZ](#).

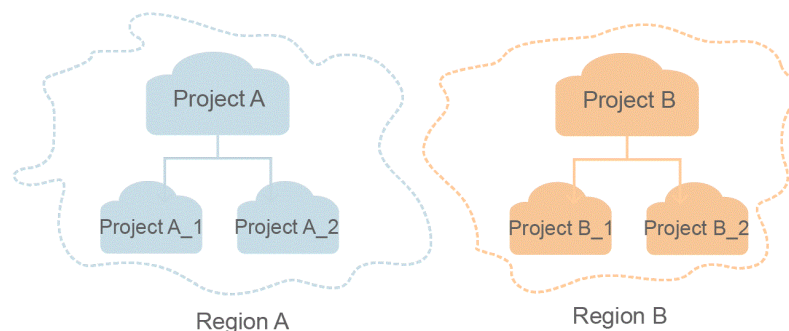
- AZ

An Availability Zone (AZ) contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to support cross-AZ high-availability systems.

- Project

Projects group and isolate resources (including compute, storage, and network resources) across physical regions. A default project is provided for each region, and subprojects can be created under each default project. Users can be granted permissions to access all resources in a specific project. If you need more refined access control, you can create subprojects under a default project and purchase resources in subprojects. Then you can assign required permissions for users to access only the resources in specific subprojects.

Figure 1-1 Project isolating model



2 API Overview

Table 2-1 RAM APIs

Type	Description
RAM Managed Permissions	List RAM managed permissions and get their details for a specified resource type.
Resource Shares	Create, list, update, or delete resource shares.
Associated Principals and Resources	Associate, disassociate, or list principals and resources for a resource share.
Associated RAM Managed Permissions	Associate or replace, disassociate, or list RAM managed permissions for a resource share.
Shared Resources	List shared resources in a resource share.
Principals	List principals in a resource share.
Resource Sharing Invitations	Accept, reject, or list resource sharing invitations.
Sharing with Organizations	Retrieve, enable, or disable sharing with Organizations.

3 Calling APIs

3.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for obtaining a user token ([Obtaining a User Token Through Password Authentication](#)) as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}

Table 3-1 Parameter description

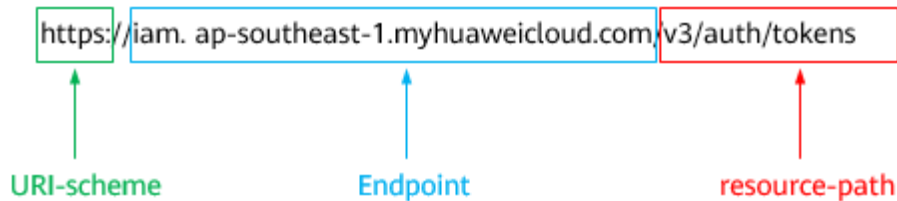
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server running the REST service. The endpoint varies between services in different regions. It can be obtained from Regions and Endpoints . For example, the endpoint of IAM in the CN-Hong Kong region is iam.ap-southeast-1.myhuaweicloud.com .
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the resource-path of the API used to obtain a user token is /v3/auth/tokens .
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of " <i>Parameter name=Parameter value</i> ". For example, ?limit=10 indicates that a maximum of 10 data records will be displayed.

For example, to obtain an IAM token in the **CN-Hong Kong** region, obtain the endpoint (`iam.ap-southeast-1.myhuaweicloud.com`) of IAM for this region and the **resource-path** (`/v3/auth/tokens`) in the URI of the API ([Obtaining a User Token Through Password Authentication](#)). Then, construct the URI as follows:

```
https://v3/auth/tokens
```

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Figure 3-1 Example URI



NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server:

- **GET**: requests the server to return specified resources.
- **PUT**: requests the server to update specified resources.
- **POST**: requests the server to add resources or perform special operations.
- **DELETE**: requests the server to delete specified resources, for example, an object.
- **HEAD**: same as GET except that the server must return only the response header.
- **PATCH**: requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API used to obtain a user token ([Obtaining a User Token Through Password Authentication](#)), the request method is POST. The request is as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows:

- **Content-Type:** specifies the request body type or format. This field is mandatory and its default value is **application/json**.
- **X-Auth-Token:** specifies a user token only for token-based API authentication. **X-Auth-Token** is a response to the API used to obtain a user token ([Obtaining a User Token Through Password Authentication](#)). This API is the only one that does not require authentication.

 **NOTE**

In addition to supporting token-based authentication, public cloud APIs also support authentication using AK/SK. During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request. For more information, see [AK/SK-based Authentication](#).

The API used to obtain a user token ([Obtaining a User Token Through Password Authentication](#)) does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

Request Body

The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to obtain a user token ([Obtaining a User Token Through Password Authentication](#)), the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace *username*, *domainname*, ******* (login password), and *xxxxxxxxxxxxxxxxxxxx* (project ID) with the actual values. To learn how to obtain a project ID, see [Obtaining Information About Account, IAM User, Group, Project, Region, and Agency](#).

 **NOTE**

The **scope** parameter specifies where a token takes effect. You can set **scope** to a project or a domain. When you set **scope** to a domain, the token takes effect for all the resources in the specified domain. In the following example, the token takes effect only for the resources in the specified project. For more details about this API, see [Obtaining a User Token Through Password Authentication](#).

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
```

```
    "domain": {
      "name": "domainname"
    }
  },
  "scope": {
    "project": {
      "id": "xxxxxxxxxxxxxxxxxxxxx"
    }
  }
}
```

If all data required for the API request is available, you can send the request to call the API through [curl](#), [Postman](#), or coding. In the response to the API used to obtain a user token ([Obtaining a User Token Through Password Authentication](#)), **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

3.2 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair.

Token-based Authentication

NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

You can obtain a token by calling the API described in [Obtaining a User Token Through Password Authentication](#). IAM APIs can be called only by using a global service token. To call the API described in [Obtaining a User Token Through Password Authentication](#), set **auth.scope** to **domain** in the request body as follows:

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "domain": {
            "name": "IAMDomain"
          },
          "name": "IAMUser",
          "password": "IAMPassword"
        }
      }
    }
  }
}
```

```
},  
  "scope": {  
    "domain": {  
      "name": "IAMDomain"  
    }  
  }  
}
```

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:

```
GET https://iam.ap-southeast-1.myhuaweicloud.com  
/v3/auth/projects  
Content-Type: application/json  
X-Auth-Token: ABCDEFJ....
```

AK/SK-based Authentication

NOTE

AK/SK-based authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token-based authentication is recommended.

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK-based authentication, you can use an AK/SK to sign requests based on the signature algorithm or use the signing SDK to sign requests. For details about how to sign requests and use the signing SDK, see [API Request Signing Guide](#).

NOTICE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

3.3 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits ranging from 1xx to 5xx. It indicates the status of a response. For more information, see [Status Codes](#).

For example, if status code 201 is returned for calling the API used to obtain a user token ([Obtaining a User Token Through Password Authentication](#)), the request is successful.

Response Header

Similar to a request, a response also has a header, for example, **Content-Type**.

Figure 1 shows the response header fields for the API used to obtain a user token (**Obtaining a User Token Through Password Authentication**). The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

Figure 3-2 Header fields of the response to the request for obtaining a user token

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token → MIIYXQVJKoZlhvcNAQcCoIIYTCCEGoCAQEExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0B8BwGgghacBIIWmHsidG9rZW4iOnsiZXhwaXJlc19hdCI6ijwMTktMDItMTNUMC
fj3KJ56YgKnpVNRbW2eZ5eb78SZOkqjACgkqlQ1wi4JlGzrpd18LGXK5tdfdq4lqHCYb8P4NaY0NYejcAgzJVeFYtLWT1GSO0zxKZmlQHQj82HBqHdglZO9fuEbL5dMhdavj+33wEI
xHRC9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXl1jipPEGA270g1FruooL6jqgIFkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUx3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbPvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==
x-xss-protection → 1; mode=block;
```

Response Body

The body of a response is often returned in structured format as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to obtain a user token (see **Obtaining a User Token Through Password Authentication**).

```
{
  "token": {
    "expires_at": "2019-02-13T06:52:13.855000Z",
    "methods": [
      "password"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "region_id": "ap-southeast-1",
            .....

```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
  "error_msg": "The format of message is error",
  "error_code": "AS.0001"
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 API

4.1 RAM Managed Permissions

4.1.1 Getting a List of RAM Managed Permissions

Function

This API is used to get a list of RAM managed permissions for the specified resource type.

URI

GET /v1/permissions

Table 4-1 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of pages that can be displayed on a page Minimum: 1 Maximum: 2000
marker	No	String	Page marker Minimum: 1 Maximum: 64
resource_type	No	String	Name of the resource type Minimum: 0 Maximum: 64

Request Parameters

None

Response Parameters

Status code: 200

Table 4-2 Response body parameters

Parameter	Type	Description
permissions	Array of PermissionSummary objects	List of details about RAM managed permissions.
page_info	PageInfo object	Pagination details

Table 4-3 PermissionSummary

Parameter	Type	Description
id	String	Permission ID Minimum: 0 Maximum: 36
name	String	Permission name Minimum: 1 Maximum: 128
resource_type	String	Resource type to which the permission applies Minimum: 0 Maximum: 100
is_resource_type_default	Boolean	Whether the permission is the default permission for the resource type
created_at	String	Time when the permission was created
updated_at	String	Time when the permission was last updated

Table 4-4 PageInfo

Parameter	Type	Description
previous_marker	String	Marker of the previous page Minimum: 1 Maximum: 64
next_marker	String	Marker of the next page Minimum: 1 Maximum: 64
current_count	Integer	Page size Minimum: 1 Maximum: 2000

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "permissions": [ {
    "id": "string",
    "name": "string",
    "resource_type": "string",
    "is_resource_type_default": true,
    "created_at": "2022-09-19T02:02:18.155Z",
    "updated_at": "2022-09-19T02:02:18.155Z"
  } ],
  "page_info": {
    "previous_marker": "string",
    "next_marker": "string",
    "current_count": 2000
  }
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.1.2 Getting Details About RAM Managed Permissions

Function

This API is used to get the details of RAM managed permissions for the specified resource type.

URI

GET /v1/permissions/{permission_id}

Table 4-5 Path Parameters

Parameter	Mandatory	Type	Description
permission_id	Yes	String	ID of the RAM managed permission Minimum: 0 Maximum: 36

Request Parameters

None

Response Parameters

Status code: 200

Table 4-6 Response body parameters

Parameter	Type	Description
permission	Permission object	Details about RAM managed permissions

Table 4-7 Permission

Parameter	Type	Description
id	String	Permission ID Minimum: 0 Maximum: 36
name	String	Permission name Minimum: 1 Maximum: 128

Parameter	Type	Description
resource_type	String	Resource type Minimum: 0 Maximum: 100
content	String	Impact and actions allowed by the permission Minimum: 1 Maximum: 1000000
is_resource_type_default	Boolean	Whether the permission is the default permission for the resource type
created_at	String	Time when the permission was created
updated_at	String	Time when the permission was last updated

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "permission" : {
    "id" : "string",
    "name" : "string",
    "resource_type" : "string",
    "content" : "string",
    "is_resource_type_default" : true,
    "created_at" : "2022-08-22T11:40:31.871Z",
    "updated_at" : "2022-08-22T11:40:31.871Z"
  }
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.2 Resource Shares

4.2.1 Creating a Resource Share

Function

This API is used to create a resource share. You can specify the list of resources to be shared, the list of principals in resource sharing, and the list of permissions granted to the principals.

URI

POST /v1/resource-shares

Request Parameters

Table 4-8 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the resource share Minimum: 1 Maximum: 64
description	No	String	Description of the resource share Minimum: 1 Maximum: 256
permission_ids	No	Array of strings	List of RAM permissions associated with the resource share. A resource type can be associated with only one RAM permission. If you do not specify a permission ID, RAM automatically associates the default permission for each resource type. Minimum: 0 Maximum: 36
principals	No	Array of strings	List of one or more principals associated with the resource share Minimum: 0 Maximum: 1024
resource_urns	No	Array of strings	List of URNs of one or more resources associated with the resource share Minimum: 0 Maximum: 1024

Response Parameters

Status code: 201

Table 4-9 Response body parameters

Parameter	Type	Description
resource_share	ResourceShare object	Details about the resource share

Table 4-10 ResourceShare

Parameter	Type	Description
id	String	ID of the resource share Minimum: 0 Maximum: 36
name	String	Name of the resource share Minimum: 0 Maximum: 128
description	String	Description of the resource share Minimum: 1 Maximum: 1024
owning_account_id	String	ID of the resource owner in a resource share Minimum: 0 Maximum: 36
status	String	Status of the resource share Minimum: 0 Maximum: 36
tags	Array of Tag objects	List of tags attached to the resource share
created_at	String	Time when the resource share was created
updated_at	String	Time when the resource share was last updated

Table 4-11 Tag

Parameter	Type	Description
key	String	Identifier or name of the tag key Minimum: 1 Maximum: 256
value	String	Tag value. You can set the tag value to an empty string, but cannot set it to null. Minimum: 0 Maximum: 1024

Example Requests

```
{
  "permissions": [{
    "id": "string",
    "name": "string",
    "resource_type": "string",
    "is_resource_type_default": true,
    "created_at": "2022-09-19T02:02:18.155Z",
    "updated_at": "2022-09-19T02:02:18.155Z"
  }],
  "page_info": {
    "previous_marker": "string",
    "next_marker": "string",
    "current_count": 2000
  }
}
```

Example Responses

Status code: 201

Request succeeded.

```
{
  "resource_share": {
    "id": "string",
    "name": "string",
    "description": "string",
    "owning_account_id": "string",
    "status": "string",
    "tags": [{
      "key": "string",
      "value": "string"
    }],
    "created_at": "2022-09-27T03:14:08.883Z",
    "updated_at": "2022-09-27T03:14:08.883Z"
  }
}
```

Status Codes

Status Code	Description
201	Request succeeded.

Error Codes

See [Error Codes](#).

4.2.2 Searching for a Resource Share

Function

This API is used to search for the details about the resource share that you have created or that shared with you.

URI

POST /v1/resource-shares/search

Request Parameters

Table 4-12 Request body parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of pages that can be displayed on a page Minimum: 1 Maximum: 2000
name	No	String	Name of the resource share Minimum: 0 Maximum: 64
marker	No	String	Page marker Minimum: 1 Maximum: 64
permission_id	No	String	Permission ID Minimum: 0 Maximum: 36

Parameter	Mandatory	Type	Description
resource_owner	Yes	String	Whether the resource share is created by you (self) or shared with you (other-accounts) Enumeration values: <ul style="list-style-type: none"> • self • other-accounts
resource_share_ids	No	Array of strings	List of resource share IDs Minimum: 0 Maximum: 36
resource_share_status	No	String	Status of the resource share Minimum: 0 Maximum: 36

Response Parameters

Status code: 200

Table 4-13 Response body parameters

Parameter	Type	Description
resource_shares	Array of ResourceShare objects	List of details about resource shares
page_info	PageInfo object	Pagination details

Table 4-14 ResourceShare

Parameter	Type	Description
id	String	ID of the resource share Minimum: 0 Maximum: 36
name	String	Name of the resource share Minimum: 0 Maximum: 128

Parameter	Type	Description
description	String	Description of the resource share Minimum: 1 Maximum: 1024
owning_account_id	String	ID of the resource owner in a resource share Minimum: 0 Maximum: 36
status	String	Status of the resource share Minimum: 0 Maximum: 36
tags	Array of Tag objects	List of tags attached to the resource share
created_at	String	Time when the resource share was created
updated_at	String	Time when the resource share was last updated

Table 4-15 Tag

Parameter	Type	Description
key	String	Identifier or name of the tag key Minimum: 1 Maximum: 256
value	String	Tag value. You can set the tag value to an empty string, but cannot set it to null. Minimum: 0 Maximum: 1024

Table 4-16 PageInfo

Parameter	Type	Description
previous_marker	String	Marker of the previous page Minimum: 1 Maximum: 64
next_marker	String	Marker of the next page Minimum: 1 Maximum: 64

Parameter	Type	Description
current_count	Integer	Page size Minimum: 1 Maximum: 2000

Example Requests

```
{
  "limit" : 2000,
  "name" : "string",
  "marker" : "string",
  "permission_id" : "string",
  "resource_owner" : "self",
  "resource_share_ids" : [ "string" ],
  "resource_share_status" : "string"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "resource_shares" : [ {
    "id" : "string",
    "name" : "string",
    "description" : "string",
    "owning_account_id" : "string",
    "status" : "string",
    "created_at" : "2022-09-19T02:06:08.590Z",
    "updated_at" : "2022-09-19T02:06:08.590Z"
  } ],
  "page_info" : {
    "previous_marker" : "string",
    "next_marker" : "string",
    "current_count" : 2000
  }
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.2.3 Updating a Resource Share

Function

This API is used to modify the properties of a resource share.

URI

PUT /v1/resource-shares/{resource_share_id}

Table 4-17 Path Parameters

Parameter	Mandatory	Type	Description
resource_share_id	Yes	String	ID of the resource share Minimum: 0 Maximum: 36

Request Parameters

Table 4-18 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the resource share Minimum: 1 Maximum: 64
description	No	String	Description of the resource share Minimum: 1 Maximum: 256

Response Parameters

Status code: 200

Table 4-19 Response body parameters

Parameter	Type	Description
resource_share	ResourceShare object	Details about the resource share

Table 4-20 ResourceShare

Parameter	Type	Description
id	String	ID of the resource share Minimum: 0 Maximum: 36
name	String	Name of the resource share Minimum: 0 Maximum: 128
description	String	Description of the resource share Minimum: 1 Maximum: 1024
owning_account_id	String	ID of the resource owner in a resource share Minimum: 0 Maximum: 36
status	String	Status of the resource share Minimum: 0 Maximum: 36
tags	Array of Tag objects	List of tags attached to the resource share
created_at	String	Time when the resource share was created
updated_at	String	Time when the resource share was last updated

Table 4-21 Tag

Parameter	Type	Description
key	String	Identifier or name of the tag key Minimum: 1 Maximum: 256
value	String	Tag value. You can set the tag value to an empty string, but cannot set it to null. Minimum: 0 Maximum: 1024

Example Requests

```
{  
  "name" : "string",
```

```
"description" : "string"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "resource_share" : {
    "id" : "string",
    "name" : "string",
    "description" : "string",
    "owning_account_id" : "string",
    "status" : "string",
    "created_at" : "2022-08-22T11:46:49.871Z",
    "updated_at" : "2022-08-22T11:46:49.871Z"
  }
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.2.4 Deleting a Resource Share

Function

This API is used to delete a specified resource share. This operation does not delete the resources but stops sharing the resources with other accounts.

URI

DELETE /v1/resource-shares/{resource_share_id}

Table 4-22 Path Parameters

Parameter	Mandatory	Type	Description
resource_share_id	Yes	String	ID of the resource share Minimum: 0 Maximum: 36

Request Parameters

None

Response Parameters

None

Example Requests

None

Example Responses

None

Status Codes

Status Code	Description
204	Request succeeded.

Error Codes

See [Error Codes](#).

4.3 Associated Principals and Resources

4.3.1 Associating Principals and Resources

Function

This API is used to associate a list of specified principals or resources with a resource share. If a resource is associated, the principals that have the permission to access the resource share will gain the permission to access that resource in the resource share. If a principal is associated, the principal will have the permission to access the shared resources in the resource share.

URI

POST /v1/resource-shares/{resource_share_id}/associate

Table 4-23 Path Parameters

Parameter	Mandatory	Type	Description
resource_share_id	Yes	String	ID of the resource share Minimum: 0 Maximum: 36

Request Parameters

Table 4-24 Request body parameters

Parameter	Mandatory	Type	Description
principals	No	Array of strings	List of one or more principals associated with the resource share Minimum: 0 Maximum: 1024
resource_urns	No	Array of strings	List of URNs of one or more resources associated with the resource share Minimum: 0 Maximum: 1024

Response Parameters

Status code: 200

Table 4-25 Response body parameters

Parameter	Type	Description
resource_share_associations	Array of ResourceShareAssociation objects	Details about the principals or resources associated with a resource share

Table 4-26 ResourceShareAssociation

Parameter	Type	Description
associated_entity	String	Associated entity. This entity can be one of the following: URN of the resource, account ID, URN of a resource directory, and URN of a folder. Minimum: 0 Maximum: 1500
association_type	String	Entity type in the association Minimum: 0 Maximum: 36
created_at	String	Time when the association was created
updated_at	String	Time when the association was last updated
resource_share_id	String	ID of the resource share Minimum: 0 Maximum: 36
status	String	Status of the association Minimum: 0 Maximum: 36

Example Requests

```
{
  "principals": [ "string" ],
  "resource_urns": [ "string" ]
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "resource_share_associations": [ {
    "associated_entity": "string",
    "association_type": "string",
    "created_at": "2022-08-22T11:50:25.639Z",
    "updated_at": "2022-08-22T11:50:25.639Z",
    "resource_share_id": "string",
    "status": "string"
  } ]
}
```


Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.3.2 Disassociating Principals and Resources

Function

This API is used to disassociate specified principals or resources from a specified resource share.

URI

POST /v1/resource-shares/{resource_share_id}/disassociate

Table 4-27 Path Parameters

Parameter	Mandatory	Type	Description
resource_share_id	Yes	String	ID of the resource share Minimum: 0 Maximum: 36

Request Parameters

Table 4-28 Request body parameters

Parameter	Mandatory	Type	Description
principals	No	Array of strings	List of one or more principals associated with the resource share Minimum: 0 Maximum: 1024
resource_urns	No	Array of strings	List of URNs of one or more resources associated with the resource share Minimum: 0 Maximum: 1024

Response Parameters

Status code: 200

Table 4-29 Response body parameters

Parameter	Type	Description
resource_share_associations	Array of ResourceShareAssociation objects	Details about the principals or resources associated with a resource share

Table 4-30 ResourceShareAssociation

Parameter	Type	Description
associated_entity	String	Associated entity. This entity can be one of the following: URN of the resource, account ID, URN of a resource directory, and URN of a folder. Minimum: 0 Maximum: 1500
association_type	String	Entity type in the association Minimum: 0 Maximum: 36
created_at	String	Time when the association was created
updated_at	String	Time when the association was last updated
resource_share_id	String	ID of the resource share Minimum: 0 Maximum: 36
status	String	Status of the association Minimum: 0 Maximum: 36

Example Requests

```
{
  "principals": [ "string" ],
  "resource_urns": [ "string" ]
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "resource_share_associations": [ {
    "associated_entity": "string",
    "association_type": "string",
    "created_at": "2022-08-22T11:53:04.912Z",
    "updated_at": "2022-08-22T11:53:04.912Z",
    "resource_share_id": "string",
    "status": "string"
  } ]
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.3.3 Searching for Associated Principals and Resources

Function

This API is used to search for the principals and resources associated with a resource share you created.

URI

POST /v1/resource-share-associations/search

Request Parameters

Table 4-31 Request body parameters

Parameter	Mandatory	Type	Description
association_status	No	String	Status of the association Minimum: 0 Maximum: 64
association_type	Yes	String	Association type (principal or resource) Enumeration values: <ul style="list-style-type: none"> • principal • resource

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of pages that can be displayed on a page Minimum: 1 Maximum: 2000
marker	No	String	Page marker Minimum: 1 Maximum: 64
principal	No	String	Principal associated with the resource share Minimum: 0 Maximum: 1024
resource_urn	No	String	URN of the resource associated with the resource share Minimum: 0 Maximum: 1024
resource_share_ids	No	Array of strings	List of resource share IDs Minimum: 0 Maximum: 36
resource_ids	No	Array of strings	List of resource IDs

Response Parameters

Status code: 200

Table 4-32 Response body parameters

Parameter	Type	Description
resource_share_associations	Array of ResourceShareAssociation objects	List of association details
page_info	PageInfo object	Pagination details

Table 4-33 ResourceShareAssociation

Parameter	Type	Description
associated_entity	String	Associated entity. This entity can be one of the following: URN of the resource, account ID, URN of a resource directory, and URN of a folder. Minimum: 0 Maximum: 1500
association_type	String	Entity type in the association Minimum: 0 Maximum: 36
created_at	String	Time when the association was created
updated_at	String	Time when the association was last updated
resource_share_id	String	ID of the resource share Minimum: 0 Maximum: 36
status	String	Status of the association Minimum: 0 Maximum: 36

Table 4-34 PageInfo

Parameter	Type	Description
previous_marker	String	Marker of the previous page Minimum: 1 Maximum: 64
next_marker	String	Marker of the next page Minimum: 1 Maximum: 64
current_count	Integer	Page size Minimum: 1 Maximum: 2000

Example Requests

```
{
  "association_status": "string",
  "association_type": "principal",
  "limit": 2000,
  "marker": "string",
```

```
"principal" : "string",  
"resource_urn" : "string",  
"resource_share_ids" : [ "string" ],  
"resource_ids" : [ "string" ]  
}
```

Example Responses

Status code: 200

Request succeeded.

```
{  
  "resource_share_associations" : [ {  
    "associated_entity" : "string",  
    "association_type" : "string",  
    "created_at" : "2022-09-19T02:10:16.960Z",  
    "updated_at" : "2022-09-19T02:10:16.960Z",  
    "resource_share_id" : "string",  
    "status" : "string"  
  } ],  
  "page_info" : {  
    "previous_marker" : "string",  
    "next_marker" : "string",  
    "current_count" : 2000  
  }  
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.4 Associated RAM Managed Permissions

4.4.1 Associating or Replacing RAM Managed Permissions

Function

This API is used to associate or replace the RAM managed permission for a resource type included in a resource share. You can have only one permission associated with each resource type in the resource share. You can associate a new RAM managed permission with a resource type only when there are currently no resources of that resource type in the resource share.

URI

POST /v1/resource-shares/{resource_share_id}/associate-permission

Table 4-35 Path Parameters

Parameter	Mandatory	Type	Description
resource_share_id	Yes	String	ID of the resource share Minimum: 0 Maximum: 36

Request Parameters

Table 4-36 Request body parameters

Parameter	Mandatory	Type	Description
permission_id	Yes	String	ID of the RAM managed permission Minimum: 0 Maximum: 36
replace	No	Boolean	Whether to use the specified permission to replace the existing permission associated with the resource share or to associate the specified permission for the current resource type with the resource share. Set this parameter to true to replace the current permission with the specified permission for the same resource type. Set this parameter to false to associate the specified permission with the current resource type. The default value is false. Only one permission can be associated with each resource type in the resource share. If the resource share already has a permission for the specified resource type and this parameter is set to false, an error is returned. This helps prevent accidental overwriting of the permission.

Response Parameters

None

Example Requests

```
{
  "permission_id" : "string",
  "replace" : true
}
```

Example Responses

None

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.4.2 Disassociating RAM Managed Permissions

Function

This API is used to disassociate RAM managed permissions from a resource share. The disassociation takes effect immediately after you call this API. You can disassociate RAM managed permissions for a resource type from a resource share only when there is no permission for that resource type in the resource share.

URI

POST /v1/resource-shares/{resource_share_id}/disassociate-permission

Table 4-37 Path Parameters

Parameter	Mandatory	Type	Description
resource_share_id	Yes	String	ID of the resource share Minimum: 0 Maximum: 36

Request Parameters

Table 4-38 Request body parameters

Parameter	Mandatory	Type	Description
permission_id	Yes	String	ID of the RAM managed permission Minimum: 1 Maximum: 36

Response Parameters

None

Example Requests

```
{  
  "permission_id" : "string"  
}
```

Example Responses

None

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.4.3 Getting Associated RAM Managed Permissions

Function

This API is used to get the details of RAM managed permissions associated with a resource share.

URI

GET /v1/resource-shares/{resource_share_id}/associated-permissions

Table 4-39 Path Parameters

Parameter	Mandatory	Type	Description
resource_share_id	Yes	String	ID of the resource share Minimum: 0 Maximum: 36

Table 4-40 Query Parameters

Parameter	Mandatory	Type	Description
permission_name	No	String	Name of the RAM managed permission Minimum: 1 Maximum: 64
limit	No	Integer	Maximum number of pages that can be displayed on a page Minimum: 1 Maximum: 2000
marker	No	String	Page marker Minimum: 1 Maximum: 64

Request Parameters

None

Response Parameters

Status code: 200

Table 4-41 Response body parameters

Parameter	Type	Description
associated_permissions	Array of AssociatedPermission objects	List of RAM managed permissions associated with the resource share
page_info	PageInfo object	Pagination details

Table 4-42 AssociatedPermission

Parameter	Type	Description
permission_id	String	Permission ID Minimum: 0 Maximum: 36
permission_name	String	Name of the RAM managed permission. Minimum: 0 Maximum: 128
resource_type	String	Resource type to which the permission applies Minimum: 0 Maximum: 100
status	String	Status of the permission Minimum: 0 Maximum: 36
created_at	String	Time when the permission was created
updated_at	String	Time when the permission was last updated

Table 4-43 PageInfo

Parameter	Type	Description
previous_marker	String	Marker of the previous page Minimum: 1 Maximum: 64
next_marker	String	Marker of the next page Minimum: 1 Maximum: 64
current_count	Integer	Page size Minimum: 1 Maximum: 2000

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "associated_permissions": [ {
    "permission_id": "string",
    "permission_name": "string",
    "resource_type": "string",
    "status": "string",
    "created_at": "2022-09-19T02:12:30.736Z",
    "updated_at": "2022-09-19T02:12:30.736Z"
  } ],
  "page_info": {
    "previous_marker": "string",
    "next_marker": "string",
    "current_count": 2000
  }
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.5 Shared Resources

4.5.1 Searching for Shared Resources

Function

This API is used to search for the resources that you share or are shared with you.

URI

POST /v1/shared-resources/search

Request Parameters

Table 4-44 Request body parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of pages that can be displayed on a page Minimum: 1 Maximum: 2000

Parameter	Mandatory	Type	Description
marker	No	String	Page marker Minimum: 1 Maximum: 64
principal	No	String	Principal associated with the resource share Minimum: 1 Maximum: 1024
resource_ids	No	Array of strings	List of resource IDs Minimum: 1 Maximum: 512
resource_urns	No	Array of strings	List of resource URNs Minimum: 0 Maximum: 1024
resource_owner	Yes	String	Resource owner (self or other-accounts) of the resource share Enumeration values: <ul style="list-style-type: none">• self• other-accounts
resource_share_ids	No	Array of strings	List of resource share IDs Minimum: 0 Maximum: 36
resource_region	No	String	Region where the resource is located Minimum: 0 Maximum: 64
resource_type	No	String	Resource type Minimum: 0 Maximum: 64

Response Parameters

Status code: 200

Table 4-45 Response body parameters

Parameter	Type	Description
shared_resources	Array of SharedResource objects	List of shared resources
page_info	PageInfo object	Pagination details

Table 4-46 SharedResource

Parameter	Type	Description
resource_urn	String	Uniform resource name of the resource Minimum: 0 Maximum: 1500
created_at	String	Time when the resource was associated with the resource share
updated_at	String	Time when the resource share was last updated
resource_type	String	Resource type Minimum: 0 Maximum: 100
resource_share_id	String	ID of the resource share associated with the resource Minimum: 0 Maximum: 36
status	String	Status of the association Minimum: 0 Maximum: 36

Table 4-47 PageInfo

Parameter	Type	Description
previous_marker	String	Marker of the previous page Minimum: 1 Maximum: 64

Parameter	Type	Description
next_marker	String	Marker of the next page Minimum: 1 Maximum: 64
current_count	Integer	Page size Minimum: 1 Maximum: 2000

Example Requests

```
{
  "limit" : 2000,
  "marker" : "string",
  "principal" : "string",
  "resource_ids" : [ "string" ],
  "resource_urns" : [ "string" ],
  "resource_owner" : "self",
  "resource_share_ids" : [ "string" ],
  "resource_region" : "string",
  "resource_type" : "string"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "shared_resources" : [ {
    "resource_urn" : "string",
    "created_at" : "2022-09-19T02:32:23.735Z",
    "updated_at" : "2022-09-19T02:32:23.735Z",
    "resource_type" : "string",
    "resource_share_id" : "string",
    "status" : "string"
  } ],
  "page_info" : {
    "previous_marker" : "string",
    "next_marker" : "string",
    "current_count" : 2000
  }
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.6 Principals

4.6.1 Searching for Principals

Function

This API is used to search for the principals that have access to shared resources.

URI

POST /v1/shared-principals/search

Request Parameters

Table 4-48 Request body parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of pages that can be displayed on a page Minimum: 1 Maximum: 2000
marker	No	String	Page marker Minimum: 1 Maximum: 64
principals	No	Array of strings	Principal associated with the resource share Maximum: 1024
resource_urn	No	String	URN of the resource Minimum: 0 Maximum: 1024
resource_owner	Yes	String	Resource owner (self or other-accounts) of the resource share Enumeration values: <ul style="list-style-type: none">• self• other-accounts
resource_share_ids	No	Array of strings	List of resource share IDs Minimum: 0 Maximum: 36

Response Parameters

Status code: 200

Table 4-49 Response body parameters

Parameter	Type	Description
shared_principals	Array of SharedPrincipal objects	List of principal details
page_info	PageInfo object	Pagination details

Table 4-50 SharedPrincipal

Parameter	Type	Description
resource_share_id	String	ID of the resource share associated with the principal Minimum: 0 Maximum: 36
id	String	Account ID of the principal or URN of the resource Minimum: 0 Maximum: 1024
created_at	String	Time when the principal was associated with the resource share
updated_at	String	Time when the resource share was last updated

Table 4-51 PageInfo

Parameter	Type	Description
previous_marker	String	Marker of the previous page Minimum: 1 Maximum: 64
next_marker	String	Marker of the next page Minimum: 1 Maximum: 64

Parameter	Type	Description
current_count	Integer	Page size Minimum: 1 Maximum: 2000

Example Requests

```
{
  "limit" : 2000,
  "marker" : "string",
  "principals" : [ "string" ],
  "resource_urn" : "string",
  "resource_owner" : "self",
  "resource_share_ids" : [ "string" ]
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "shared_principals" : [ {
    "resource_share_id" : "string",
    "id" : "string",
    "created_at" : "2022-09-19T02:33:13.496Z",
    "updated_at" : "2022-09-19T02:33:13.496Z"
  } ],
  "page_info" : {
    "previous_marker" : "string",
    "next_marker" : "string",
    "current_count" : 2000
  }
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.7 Resource Sharing Invitations

4.7.1 Accepting a Resource Sharing Invitation

Function

This API is used to accept a resource sharing invitation from other accounts.

URI

POST /v1/resource-share-invitations/{resource_share_invitation_id}/accept

Table 4-52 Path Parameters

Parameter	Mandatory	Type	Description
resource_share_invitation_id	Yes	String	ID of the resource sharing invitation Minimum: 0 Maximum: 36

Request Parameters

None

Response Parameters

Status code: 200

Table 4-53 Response body parameters

Parameter	Type	Description
resource_share_invitation	ResourceShareInvitation object	Details about the resource sharing invitation

Table 4-54 ResourceShareInvitation

Parameter	Type	Description
receiver_account_id	String	ID of the account that receives the resource sharing invitation Minimum: 0 Maximum: 36
resource_share_id	String	ID of the resource share Minimum: 0 Maximum: 36

Parameter	Type	Description
resource_share_name	String	Name of the resource share Minimum: 0 Maximum: 64
resource_share_invitation_id	String	ID of the resource sharing invitation Minimum: 0 Maximum: 36
sender_account_id	String	ID of the account that sends a resource share invitation Minimum: 0 Maximum: 36
status	String	Status of the resource sharing invitation Minimum: 0 Maximum: 64
created_at	String	Time when the invitation was created
updated_at	String	Time when the invitation was last updated

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "resource_share_invitation" : {
    "receiver_account_id" : "string",
    "resource_share_id" : "string",
    "resource_share_name" : "string",
    "resource_share_invitation_id" : "string",
    "sender_account_id" : "string",
    "status" : "string",
    "created_at" : "2022-08-22T12:28:26.631Z",
    "updated_at" : "2022-08-22T12:28:26.631Z"
  }
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.7.2 Rejecting a Resource Sharing Invitation

Function

This API is used to reject a resource sharing invitation from other accounts.

URI

POST /v1/resource-share-invitations/{resource_share_invitation_id}/reject

Table 4-55 Path Parameters

Parameter	Mandatory	Type	Description
resource_share_invitation_id	Yes	String	ID of the resource sharing invitation Minimum: 0 Maximum: 36

Request Parameters

None

Response Parameters

Status code: 200

Table 4-56 Response body parameters

Parameter	Type	Description
resource_share_invitation	ResourceShareInvitation object	Details about the resource sharing invitation

Table 4-57 ResourceShareInvitation

Parameter	Type	Description
receiver_account_id	String	ID of the account that receives the resource sharing invitation Minimum: 0 Maximum: 36

Parameter	Type	Description
resource_share_id	String	ID of the resource share Minimum: 0 Maximum: 36
resource_share_name	String	Name of the resource share Minimum: 0 Maximum: 64
resource_share_invitation_id	String	ID of the resource sharing invitation Minimum: 0 Maximum: 36
sender_account_id	String	ID of the account that sends a resource share invitation Minimum: 0 Maximum: 36
status	String	Status of the resource sharing invitation Minimum: 0 Maximum: 64
created_at	String	Time when the invitation was created
updated_at	String	Time when the invitation was last updated

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "resource_share_invitation": {
    "receiver_account_id": "string",
    "resource_share_id": "string",
    "resource_share_name": "string",
    "resource_share_invitation_id": "string",
    "sender_account_id": "string",
    "status": "string",
    "created_at": "2022-08-22T12:29:00.460Z",
    "updated_at": "2022-08-22T12:29:00.460Z"
  }
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.7.3 Searching for a Resource Sharing Invitation

Function

This API is used to search for a resource sharing invitation based on certain search criteria.

URI

POST /v1/resource-share-invitations/search

Request Parameters

Table 4-58 Request body parameters

Parameter	Mandatory	Type	Description
resource_share_ids	No	Array of strings	List of resource share IDs Minimum: 0 Maximum: 36
resource_share_invitation_ids	No	Array of strings	List of resource sharing invitation IDs Minimum: 0 Maximum: 36
status	No	String	Status of the resource sharing invitation Minimum: 0 Maximum: 64
limit	No	Integer	Maximum number of pages that can be displayed on a page Minimum: 1 Maximum: 2000

Parameter	Mandatory	Type	Description
marker	No	String	Paging location marker. The query starts from the next piece of data indexed by this parameter. You do not need to specify this parameter when you query resources on the first page. When you query resources on subsequent pages, set this parameter to the location returned in the response body for the previous query. Minimum: 1 Maximum: 64

Response Parameters

Status code: 200

Table 4-59 Response body parameters

Parameter	Type	Description
resource_share_invitations	Array of ResourceShareInvitation objects	Details about the resource sharing invitation
page_info	PageInfo object	Pagination details

Table 4-60 ResourceShareInvitation

Parameter	Type	Description
receiver_account_id	String	ID of the account that receives the resource sharing invitation Minimum: 0 Maximum: 36
resource_share_id	String	ID of the resource share Minimum: 0 Maximum: 36

Parameter	Type	Description
resource_share_name	String	Name of the resource share Minimum: 0 Maximum: 64
resource_share_invitation_id	String	ID of the resource sharing invitation Minimum: 0 Maximum: 36
sender_account_id	String	ID of the account that sends a resource share invitation Minimum: 0 Maximum: 36
status	String	Status of the resource sharing invitation Minimum: 0 Maximum: 64
created_at	String	Time when the invitation was created
updated_at	String	Time when the invitation was last updated

Table 4-61 PageInfo

Parameter	Type	Description
previous_marker	String	Marker of the previous page Minimum: 1 Maximum: 64
next_marker	String	Marker of the next page Minimum: 1 Maximum: 64
current_count	Integer	Page size Minimum: 1 Maximum: 2000

Example Requests

```
{
  "resource_share_ids" : [ "string" ],
  "resource_share_invitation_ids" : [ "string" ],
  "status" : "string",
  "limit" : 2000,
  "marker" : "string"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "resource_share_invitations": [ {
    "receiver_account_id": "string",
    "resource_share_id": "string",
    "resource_share_name": "string",
    "resource_share_invitation_id": "string",
    "sender_account_id": "string",
    "status": "string",
    "created_at": "2022-09-23T06:43:43.999Z",
    "updated_at": "2022-09-23T06:43:43.999Z"
  } ],
  "page_info": {
    "previous_marker": "string",
    "next_marker": "string",
    "current_count": 2000
  }
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.8 Sharing with Organizations

4.8.1 Checking Whether Sharing with Organizations Is Enabled

Function

This API is used to check whether sharing with Organizations is enabled.

URI

GET /v1/organization-share

Request Parameters

None

Response Parameters

Status code: 200

Table 4-62 Response body parameters

Parameter	Type	Description
enabled	Boolean	Whether sharing with Organizations is enabled. If the value is true, sharing with Organizations is enabled. The default value is false.

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{  
  "enabled" : true  
}
```

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.8.2 Enabling Sharing with Organizations

Function

This API is used to enable sharing with Organizations.

URI

POST /v1/organization-share/enable

Request Parameters

None

Response Parameters

None

Example Requests

None

Example Responses

None

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

4.8.3 Disabling Sharing with Organizations

Function

This API is used to disable sharing with Organizations.

URI

POST /v1/organization-share/disable

Request Parameters

None

Response Parameters

None

Example Requests

None

Example Responses

None

Status Codes

Status Code	Description
200	Request succeeded.

Error Codes

See [Error Codes](#).

5 Permissions Policies and Supported Actions

5.1 Introduction

This section describes fine-grained permissions management for your RAM. If your account does not need individual Identity and Access Management (IAM) users, then you may skip over this section.

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specific operations on cloud services based on the assigned permissions.

You can grant users permissions by using [roles](#) and [policies](#). Roles are provided by IAM to define service-based permissions depending on user's job responsibilities. Policies define API-based permissions for operations on specific resources under certain conditions, allowing for more fine-grained, secure access control of cloud resources.

NOTE

Use policy-based authorization if you want to allow or deny the access to an API.

An account has all the permissions required to call all APIs, but IAM users must be assigned the required permissions to do so. The permissions required for calling an API are determined by the actions supported by the API. Only users who have been granted permissions to perform the actions can call the API successfully. For example, if an IAM user wants to query RAM using an API, the user must have been granted permissions that allow the **ram:permissions:list** action.

Supported Actions

There are system-defined policies that can be directly used. You can also create custom policies and use them to supplement system-defined policies for more refined access control. Operations supported by policies are specific to APIs. The following are common concepts related to policies:

- Permissions: Statements in a policy that allow or deny certain operations.
- APIs: APIs that can be called in a custom policy.
- Actions: Specific operations that are allowed or denied in a custom policy.
- Dependent actions: When assigning an action to users, you also need to assign dependent permissions for that action to take effect.
- IAM projects/Enterprise projects: Authorization scope of a custom policy. A custom policy can be applied to IAM projects or enterprise projects or both. For example, if you set the authorization scope of a custom policy to both IAM projects and enterprise projects, the policy takes effect for user groups in either IAM or enterprise projects. If the authorization scope is set to IAM projects only, the custom policy will take effect only for user groups in IAM projects. For details about the differences between IAM and enterprise projects, see [Differences Between IAM and Enterprise Projects](#).

 **NOTE**

The check mark (✓) indicates that an action takes effect. The cross mark (x) indicates that an action does not take effect.

5.2 Actions

Table 5-1 RAM actions

Permission	API	Action	IAM Project	Enterprise Project
Listing RAM managed permissions	GET /v1/permissions	ram:permissions:list	×	×
Getting the details about RAM managed permissions	GET /v1/permissions/{permission_id}	ram:permissions:get	×	×
Creating a resource share	POST /v1/resource-shares	ram:resourceShares:create	×	×
Getting details about a resource share	POST /v1/resource-shares/search	ram:resourceShares:search	×	×

Permission	API	Action	IAM Project	Enterprise Project
Updating properties of a resource share	PATCH /v1/resource-shares/{resource_share_id}	ram:resourceShares:update	×	×
Deleting a resource share	DELETE /v1/resource-shares/{resource_share_id}	ram:resourceShares:delete	×	×
Associating principals and resources with a resource share	POST /v1/resource-shares/{resource_share_id}/associate	ram:resourceShares:associate	×	×
Disassociating principals and resources from a resource share	POST /v1/resource-shares/{resource_share_id}/disassociate	ram:resourceShares:disassociate	×	×
Searching for associated principals and resources	POST /v1/resource-share-associations/search	ram:resourceShares:searchResourceShareAssociations	×	×
Associating or replacing RAM managed permissions	POST /v1/resource-shares/{resource_share_id}/associate-permission	ram:resourceShares:associatePermission	×	×
Disassociating RAM managed permissions	POST /v1/resource-shares/{resource_share_id}/disassociate-permission	ram:resourceShares:disassociatePermission	×	×

Permission	API	Action	IAM Project	Enterprise Project
Listing associated RAM managed permissions	GET /v1/resource-shares/{resource_share_id}/associated-permissions	ram:resourceShares:listAssociatedPermissions	×	×
Searching for shared resources	POST /v1/shared-resources/search	ram:sharedResources:search	×	×
Searching for principals	POST /v1/shared-principals/search	ram:sharedPrincipals:search	×	×
Accepting a resource sharing invitation	POST /v1/resource-share-invitations/{resource_share_invitation_id}/accept	ram:resourceShareInvitations:accept	×	×
Rejecting a resource sharing invitation	POST /v1/resource-share-invitations/{resource_share_invitation_id}/reject	ram:resourceShareInvitations:reject	×	×
Searching for a resource sharing invitation	POST /v1/resource-share-invitations/search	ram:resourceShareInvitations:search	×	×
Enabling sharing with Organizations	POST /v1/resource-shares/enable-sharing-with-organization	ram:resourceShares:enableSharingWithOrganization	×	×
Disabling sharing with Organizations	POST /v1/resource-shares/disable-sharing-with-organization	ram:resourceShares:disableSharingWithOrganization	×	×
Checking whether sharing with Organizations is enabled	POST /v1/resource-shares/search-enabled-sharing-with-organization	ram:resourceShares:searchEnableSharingWithOrganization	×	×

6 Appendixes

6.1 Status Codes

Table 6-1 Status codes

Status Code	Message Title	Description
100	Continue	The client should continue with its request. This interim response is used to inform the client that the initial part of the request has been received and has not yet been rejected by the server.
101	Switching Protocols	The requester has asked the server to switch protocols and the server has agreed to do so. The protocol should be switched only when it is advantageous to do so. For example, switching to a newer version of HTTP is advantageous over older versions.
201	Created	The request has been fulfilled and resulted in a new resource being created.
202	Accepted	The request has been accepted for processing, but the processing has not been completed.
203	Non-Authoritative Information	The server successfully processed the request, but is returning information that may be from another source.
204	NoContent	The server successfully processed the request and is not returning any content. The status code is returned in response to an HTTP OPTIONS request.

Status Code	Message Title	Description
205	Reset Content	The server successfully processed the request, but is not returning any content.
206	Partial Content	The server has fulfilled the partial GET request for the resource.
300	Multiple Choices	There are multiple options for the resource from which the client may choose. For example, this code could be used to present a list of resource characteristics and addresses from which the client such as a browser may choose.
301	Moved Permanently	The requested resource has been assigned a new permanent URI and any future references to this resource should use one of the returned URIs.
302	Found	The requested resource resides temporarily under a different URI.
303	See Other	The response to the request can be found under a different URI and should be retrieved using a GET or POST method.
304	Not Modified	The requested resource has not been modified. When the server returns this status code, it does not return any resources.
305	Use Proxy	The requested resource must be accessed through a proxy.
306	Unused	This HTTP status code is no longer used.
400	BadRequest	The request could not be understood by the server due to malformed syntax. The client should not repeat the request without modifications.
401	Unauthorized	The authorization information provided by the client is incorrect or invalid. Check the username and password.
402	Payment Required	This status code is reserved for future use.
403	Forbidden	The server understood the request, but is refusing to fulfill it. The client should not repeat the request without modifications.
404	NotFound	The requested resource cannot be found. The client should not repeat the request without modifications.

Status Code	Message Title	Description
405	MethodNotAllowed	The method specified in the request is not allowed for the requested resource. The client should not repeat the request without modifications.
406	Not Acceptable	The server cannot fulfill the request based on the content characteristics of the request.
407	Proxy Authentication Required	This code is similar to 401, but indicates that the client must first authenticate itself with the proxy.
408	Request Time-out	The client does not produce a request within the time that the server was prepared to wait. The client may repeat the request without modifications at any later time.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource. This status code indicates that the resource that the client attempts to create already exists, or the request fails to be processed because of the update of the conflict request.
410	Gone	The requested resource is no longer available. The requested resource has been deleted permanently.
411	Length Required	The server refuses to process the request without a defined Content-Length.
412	Precondition Failed	The server does not meet one of the preconditions that the requester puts on the request.
413	Request Entity Too Large	The server is refusing to process a request because the request entity is larger than the server is willing or able to process. The server may close the connection to prevent the client from continuing the request. If the condition is temporary, the server should include a Retry-After header field to indicate that it is temporary and after what time the client may try again.
414	Request-URI Too Large	The server is refusing to service the request because the request URI is longer than the server is willing to interpret.

Status Code	Message Title	Description
415	Unsupported Media Type	The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.
416	Requested range not satisfiable	The requested range is invalid.
417	Expectation Failed	The server fails to meet the requirements of the Expect request header field.
422	UnprocessableEntity	The request was well-formed but was unable to be followed due to semantic errors.
429	TooManyRequests	The client has sent more requests than its rate limit is allowed within a given amount of time, or the server has received more requests than it is able to process within a given amount of time. In this case, the client should repeat requests after the time specified in the Retry-After header of the response expires.
500	InternalServerError	The server encountered an unexpected condition which prevented it from fulfilling the request.
501	Not Implemented	The server does not support the functionality required to fulfill the request.
502	Bad Gateway	The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.
503	ServiceUnavailable	The requested service is unavailable. The client should not repeat the request without modifications.
504	ServerTimeout	The request cannot be fulfilled within a given amount of time. The response will reach the client only if the request carries a timeout parameter.
505	HTTP Version not supported	The server does not support the HTTP protocol version used in the request.

6.2 Error Codes

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in [API Gateway Error Codes](#).

Status Code	Error Code	Error Message	Description	Solution
400	RAM.1001	The domain id in the header or the token is invalid.	Invalid account ID or token.	Check whether the account ID is correct.
400	RAM.1002	The domain id %s in the header is invalid.	Invalid account ID.	Check whether the account ID is correct.
400	RAM.1003	The token is invalid.	Invalid token.	Contact technical support.
400	RAM.1004	The account id is invalid.	Invalid account ID.	Check whether the account ID of the principal is correct.
400	RAM.1005	The share principal %s cannot be the owner.	Resource owner not allowed to act as principal.	Check whether the principal is correctly configured.
400	RAM.1006	The principal is invalid for duplicate.	Duplicate principals.	Delete the duplicate principals.
400	RAM.1007	The resource urn is invalid for duplicate.	Duplicate URNs of shared resources.	Delete the duplicate URNs.
400	RAM.1008	Bad request for checking permission.	No permissions for the shared resource.	Check whether the RAM permission has been granted to the IAM user.
400	RAM.1009	The resource type %s has no permission.	No permissions for the shared resource.	Grant the permissions to access the shared resource.
400	RAM.1010	The domain id %s does not own the resource urn %s.	Account ID and URN mismatched.	Check whether the account ID matches the URN.

Status Code	Error Code	Error Message	Description	Solution
400	RAM.1011	The %s num exceeds %s quota %s if add count %s.	Maximum quota exceeded.	Check the request parameters.
400	RAM.1012	The %s num exceeds the total quota %s if add count %s.	Maximum quota exceeded.	Check the request parameters.
400	RAM.1013	The organization management id %s has not enabled sharing.	Sharing with Organizations not enabled.	Enable sharing with Organizations.
400	RAM.1014	The cross organizations share cannot shared to the organizations with the principal %s.	Invalid cross-organization sharing.	Check whether the account of the principal belongs to another organization.
400	RAM.1015	The cross organizations share cannot shared to the organizations unit with the principal %s.	Invalid cross-organization sharing.	Check whether the account of the principal belongs to another organization.
400	RAM.1016	The cross organizations share cannot shared to the organizations root with the principal %s.	Invalid cross-organization sharing.	Check whether the account of the principal belongs to another organization.
404	RAM.1017	The resource share id %s is not found.	Invalid ID of a resource share.	Check whether the ID of the resource share is correct.

Status Code	Error Code	Error Message	Description	Solution
404	RAM.1018	The permission id %s is not found.	Invalid ID of the permission to access the shared resources.	Check whether the permission ID is correct.
404	RAM.1020	Get project failed.	Failed to get project information.	Check whether the account ID is correct.
404	RAM.1021	Get domain information failed.	Failed to obtain domain information.	Contact technical support.
404	RAM.1022	The principal %s is not exist.	Principal not found.	Check whether the principal is correctly configured.
404	RAM.1023	The principal %s is invalid.	Invalid principal.	Check whether the principal is correctly configured.
404	RAM.1024	The urn %s is invalid.	Invalid URN of the shared resources.	Check whether the URN is correct.
400	RAM.1101	The status %s of the resource share with id %s is not active.	Resource share not activated.	Check the status of the resource share.
400	RAM.1102	There are resources in use in the resource share %s.	Resource already associated.	Check whether the resource has been already associated with another resource share.
400	RAM.1103	Some resources do not have or have multiple corresponding permissions.	No permissions or multiple permissions for the resource share.	Check whether the permissions are configured correctly for the resource share.

Status Code	Error Code	Error Message	Description	Solution
400	RAM.1201	The request body is empty.	Request parameters left blank.	Check the request parameters.
409	RAM.1202	Some principals have been associated to the resource share with id %s.	Same principal already associated.	Check whether the same principal is associated with multiple resource shares.
409	RAM.1203	Some resources have been associated to the resource share with id %s.	Resource already associated.	Check whether the same resource is associated with multiple resource shares.
400	RAM.1204	The status %s of the resource share is not active.	Resource share not activated.	Contact technical support.
400	RAM.1205	Get organization id error.	Failed to obtain the organization ID.	Check whether the account ID is correct.
400	RAM.1206	Get ancestors error.	Failed to obtain the upper-level organization.	Check whether the account ID is correct.
400	RAM.1207	Unable to disassociate the principal or the resource which is not exist or owned by you.	Disassociation not allowed for the resource or principal not in your organization or that does not exist.	Check whether the principal or resource is correctly configured.
400	RAM.1208	Unable to disassociate the principal because you belong to the organization.	Disassociation not allowed for the principal in your organization.	Check whether the principal belongs to your organization.

Status Code	Error Code	Error Message	Description	Solution
400	RAM.1301	The status %s of the resource share is not active.	Resource share not activated.	Contact technical support.
409	RAM.1302	The resource type %s has associated one permission.	Permission already associated with to the shared resource.	Check the request parameters.
409	RAM.1303	The resource type %s is in use in the resource share with id %s.	Failed to disassociate the permission from the in-use shared resource.	Stop using the shared resource and then disassociate the permission from the shared resource.
409	RAM.1701	The current status of the invitation is %s.	Abnormal status of the resource sharing invitation.	Check whether the status of the resource sharing invitation is correct.
404	RAM.1702	The resource share invitation with id %s is not found.	Invalid resource sharing invitation.	Check whether the resource sharing invitation has expired.
400	RAM.1801	The account id %s is not the management account of organization.	Account ID not being the management account.	Contact the organization administrator. Only the management account can perform this operation.
400	RAM.1802	Get organization info error.	Incorrect organization information.	Check whether organization information is correct.

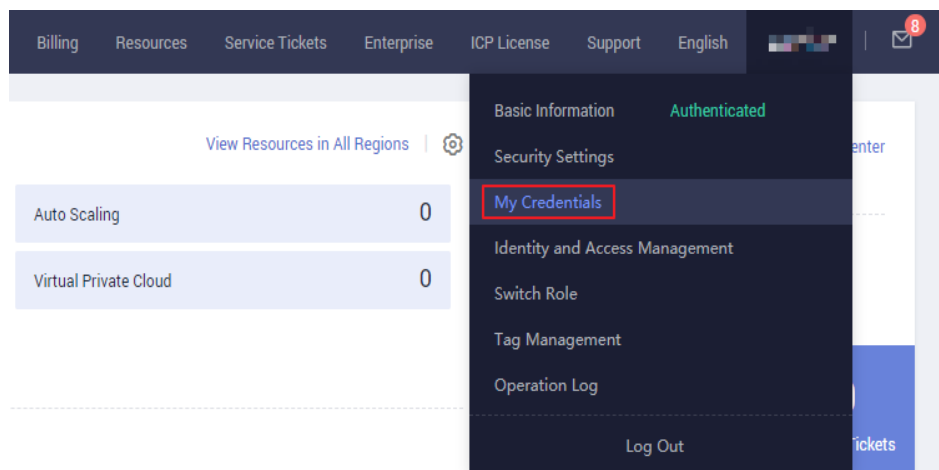
Status Code	Error Code	Error Message	Description	Solution
400	RAM.1803	The sharing with organization is not enabled by management account %s.	Sharing with Organizations not enabled by the management account.	Contact the organization administrator to use the management account to enable sharing with Organizations.
501	RAM.1804	RAM is not support organization sharing function.	Sharing with Organizations not supported by RAM.	Contact technical support.

6.3 Obtaining Information About Account, IAM User, Group, Project, Region, and Agency

Obtaining Account, IAM User, and Project Information

- Using the console
 - a. On the HUAWEI CLOUD homepage, click **Console** in the upper right corner.
 - b. Hover over the username in the upper right corner and choose **My Credentials**.

Figure 6-1 My Credentials



- c. View the account name, account ID, username, user ID, project name, and project ID on the **API Credentials** page.

7 Change History

Release On	Description
2022-10-29	The issue is the first official release.