Identity and Access Management

API Reference

Issue 15

Date 2024-04-15

HUAWEI

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Before You Start 1
1.1 Overview 1
1.2 API Calling 1
1.3 Endpoints 1
1.4 Constraints 1
1.5 Concepts 2
2 API Overview 4
3 Calling APIs 5
3.1 Making an API Request 5
3.2 Authentication 8
3.3 Response 9
4 APIs 11
4.1 Token Management 11
4.1.1 Obtaining a User Token 11
4.2 Project Management 19
4.2.1 Querying Project Information Based on the Specified Criteria 20
5 Permissions Policies and Supported Actions 23
5.1 Introduction 23
6 Appendix 25
6.1 Status Codes 25
6.2 Error Codes 29
6.3 Obtaining User, Account, User Group, Project, and Agency Information 43
A Change History 44

1 Before You Start

1.1 Overview

Welcome to Identity and Access Management (IAM). IAM provides identity authentication, permissions management, and access control. With IAM, you can create and manage users and grant them permissions to allow or deny their access to cloud resources.

You can use IAM through the console or application programming interfaces (APIs). This document describes how to use APIs to perform operations on IAM, such as creating users and user groups and obtaining tokens.

1.2 API Calling

IAM supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, 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 endpoints of all services, see **Regions and Endpoints**.

1.4 Constraints

The number of IAM resources that you can create is determined by your quota. For details, see "Notes and Constraints" in the *Identity and Access Management User Guide*.

All APIs of IAM can be called using the global region endpoint. Some APIs can be called using endpoints of both the global region and other regions (see **Table 1-1**), and other APIs can be called using only the global region endpoint.

NOTE

Tokens or temporary AKs/SKs obtained using domain names of all regions except the global region can only be used to access services in the same region.

Table 1-1 Global and region-specific APIs

Category	API URI	Link
Token Management	POST /v3/auth/tokens	**Obtaining a User Token**

1.5 Concepts

Common concepts used when you call IAM APIs are described as follows:

- Domain

  A domain, also called an "account", is created upon successful registration. The domain has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions.

- User

  A user is created using a domain to use cloud services. Each user has their own identity credentials (password and access keys).

  An IAM user can view the domain 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

  A region contains a physical data center, which is completely isolated to improve fault tolerance and stability. The region that is selected during resource creation cannot be changed after the resource is created. Regions are classified into universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

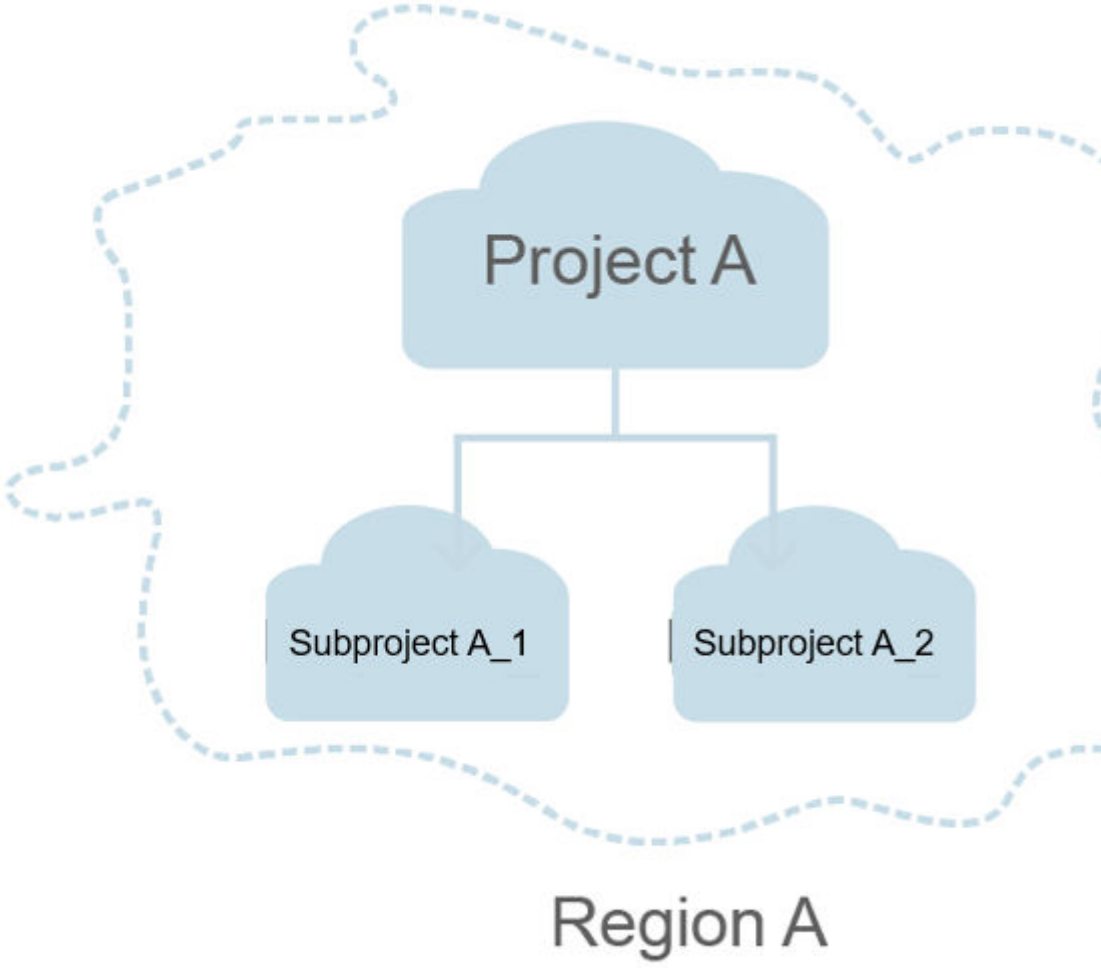
- AZ

  An AZ is a physical location where resources use independent power supplies and networks. A region contains one or more AZs that are physically isolated but interconnected through internal networks. Because AZs are isolated from each other, any fault that occurs in an AZ will not affect other AZs.

- 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. For more refined access control, create subprojects under a project and create resources in the subprojects. Users can then be assigned permissions to access only specific resources in the subprojects.

Figure 1-1 Project isolating model



2 API Overview

Token Management

API	Description
Obtaining a User Token	Obtain a user token through username/password-based authentication.

Project Management

API	Description
Querying Project Information Based on the Specified Criteria	Query project information.

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 (see **Obtaining a User Token**) 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 bearing the REST service. The endpoint varies between services in different regions.
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 "*Parameter name*=*Parameter value*". For example, **?limit=10** indicates that a maximum of 10 data records will be displayed.

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 a 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**), the request method is POST. The request is as follows:

```
POST https://{{endpoint}}/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**). This API is the only one that does not require authentication.

  NOTE

  In addition to supporting token-based authentication, APIs also support authentication using access key ID/secret access key (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 details, see "AK/SK-based Authentication" in **Authentication**.

The API used to obtain a user token (**Obtaining a User Token**) does not require authentication. 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://{{endpoint}}/v3/auth/tokens

Content-Type: application/json
```

(Optional) 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**), 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 *xxxxxxxxxxxxxxxxx* (project ID) with the actual values. If you obtain a token using an account, ensure that you set *username* and *domainname* to the same value.

NOTE

The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under the account. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see **Obtaining a User Token**.

```
POST https://{{endpoint}}/v3/auth/tokens

Content-Type: application/json

{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "id": "xxxxxxxxxxxxxxxxx"
            }
        }
    }
}
```

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**), **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 request headers to get permissions for calling the API.

You can obtain a token by calling the API described in **Obtaining a User Token**. IAM APIs can be called only by using a global service token. To call the API described in **Obtaining a User Token**, 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:

```
POST https://{{endpoint}}/v3/auth/tokens
```


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 pair to sign requests based on the signature algorithm or use the signing SDK to sign requests.

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 the 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 request. 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**), the request is successful.

Response Header

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

Obtaining a User Token shows the response header fields for the API used to obtain a user token (**Figure 3-1**). 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 3-1 Header fields of the response to the request for obtaining a user token

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 (**Obtaining a User Token**).

```
{
    "token": {
        "expires_at": "2019-02-13T06:52:13.855000Z",
        "methods": [
            "password"
        ],
        "catalog": [
            {
                "endpoints": [
                    {
                        "region_id": "az-01",
......
```

If an error occurs during API calling, an error code and error description 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 APIs

4.1 Token Management

4.1.1 Obtaining a User Token

Function

This API is used to obtain a token through username/password authentication. A token is a system object encapsulating the identity and permissions of a user. When calling the APIs of IAM or other cloud services, you can use this API to obtain a token for authentication.

NOTE

Tokens are valid for 24 hours and you can cache them to reduce the number of API calls needed. Ensure that the token is valid while you use it. Using a token that will soon expire may cause API calling failures. Obtaining a new token does not affect the validity of the existing token. The following operations will invalidate the existing token. After these operations are performed, obtain a new token.

- Changing the password or access key of your account or an IAM user: The token of your account or the user is invalidated.
- Deleting or disabling an IAM user: The token of the user is invalidated.
- Changing the permissions of an IAM user: The token of the user is invalidated. For example, when the user is added to or removed from a user group, or when permissions of the group which the user belongs to are modified.

URI

POST /v3/auth/tokens

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Fill **application/json;charset=utf8** in this field.

- Parameters in the request body

Parameter	Mandatory	Type	Description

identity	Yes	JSON object	Authentication parameters, including: **methods** and **password**.
			"identity": {
			"methods": ["password"],
			"password": {

methods	Yes	String Array	Authentication method. The value of this field is **password**. If virtual MFA–based login authentication is enabled, the value of this field is **["password","totp"]**.

password	Yes	JSON object	Authentication information. Example:
			"password": {
			"user": {
			"name": "*user A*",
			"password": "**********",
			"domain": {
			"name": "*domain A*"
			• **user.name**: Name of the user that wants to obtain the token. Obtain the username on the **My Credentials** page.
			• **password**: Login password of the user.
			• **domain.name**: Name of the domain that created the user. Obtain the domain name on the **My Credentials** page.

Parameter	Mandatory	Type	Description

totp	No	JSON object	Authentication information. This parameter is mandatory only when virtual MFA–based login authentication is enabled.
			Example:
			"totp": {
			"user": {
			"id": "b95b78b67fa045b38104c12fb...",
			"passcode": "******"
			• **user.id**: User ID, which can be obtained on the **My Credentials** page.
			• **passcode**: Virtual MFA device verification code, which can be obtained on the MFA app.

scope	No	JSON object	Usage scope of the token. The value can be **project** or **domain**.
			• Example 1: If this field is set to **project**, the token can be used to access only services in specific projects, such as ECS. You can specify either **id** or **name**.
			"scope": {
			"project": {
			"id": "0b95b78b67fa045b38104c12fb..."
			}
			}
			• Example 2: If this field is set to **domain**, the token can be used to access global services, such as OBS. Global services are not subject to any projects or regions. You can specify either **id** or **name**.
			"scope": {
			"domain": {
			"name": " domain A"
			}
			}

- Example request

  The following is a sample request for obtaining a token for **user A**. The login password of the user is ************** and the domain name is **domain A**. The scope of the token is **domain**.

```
{
  "auth": {
    "identity": {
      "methods": ["password"],
```



```
    "password": {
      "user": {
        "name": "user A",
        "password": "**********",
        "domain": {
          "name": "domain A"
        }
      }
    }
  },
  "scope": {
    "domain": {
      "name": "domain A"
    }
  }
 }
}
```

The following is a sample request for obtaining a token when virtual MFA–based login authentication is enabled.

```
{
    "auth": {
        "identity": {
            "methods": ["password", "totp"],
            "password": {
                "user": {
                    "name": "user A",
                    "password": "********",
                    "domain": {
                        "name": "domain A"
                    }
                }
            },
            "totp" : {
                "user": {
                    "id": "dfsafdfsaf....",
                    "passcode": "******"
                }
            }
        },
        "scope": {
            "domain": {
                "name": "domain A"
            }
        }
    }
}
```

Response Parameters

- Parameters in the response header

Parameter	Mandatory	Type	Description
X-Subject-Token	Yes	String	Obtained token.

- Token format description

Parameter	Mandatory	Type	Description
methods	Yes	Json Array	Method for obtaining a token.

Parameter	Mandatory	Type	Description
expires_at	Yes	String	Expiration date of the token.
issued_at	Yes	String	Time when the token was issued.
mfa_authn_at	No	String	MFA authentication time. This field is displayed only when virtual MFA–based login authentication is enabled.

user	Yes	JSON object	Example:
			"user": {
			"name": "user A",
			"id": "b95b78b67fa045b38104...",
			"password_expires_at":"2016-11-06T15:32:17.000000",
			"domain": {
			"name": "domain A",
			"id": "fdec73ffea524aa1b373e40..."
			}
			}
			• **user.name**: Name of the user that wants to obtain the token.
			• **user.id**: ID of the user.
			• **domain.name**: Name of the domain that created the user.
			• **domain.id**: ID of the domain.
			• **password_expires_at**: Coordinated Universal Time (UTC) that the password will expire. **null** indicates that the password will not expire.

Parameter	Mandatory	Type	Description
domain	No	JSON object	This parameter is returned only when the **scope** parameter in the request body has been set to **domain**. Example: `"domain": { "name" : "domain A" "id" : "fdec73ffea524aa1b373e40..."` • **domain.name**: Name of the domain that created the user. • **domain.id**: ID of the domain.
project	No	JSON object	This parameter is returned only when the **scope** parameter in the request body has been set to **project**. Example: `"project": { "name": "project A", "id": "34c77f3eaf84c00aaf54...", "domain": { "name": "domain A", "id": "fdec73ffea524aa1b373e40..." } }` • **project.name**: Name of a project. • **project.id**: ID of the project. • **domain.name**: Domain name of the project. • **domain.id**: Domain ID of the project.

Parameter	Mandatory	Type	Description

catalog	Yes	Json Array	Endpoint information.
			Example:
			"catalog": [{
			"type": "identity",
			"id": "1331e5cff2a74d76b03da1225910e...",
			"name": "iam",
			"endpoints": [{
			"url": "*https://sample.domain.com*/v3",
			"region": "*",
			"region_id": "*",
			"interface": "public",
			"id": "089d4a381d574308a703122d3ae73..."
			}]
			}]
			• **type**: Type of the service which the API belongs to.
			• **id**: ID of the service.
			• **name**: Name of the service.
			• **endpoints**: Endpoints that can be used to call the API.
			• **url**: URL used to call the API.
			• **region**: Region in which the service can be accessed.
			• **region_id**: ID of the region.
			• **interface**: Type of the API. The value **public** means that the API is open for access.
			• **id**: ID of the API.

roles	Yes	JSON object	Permissions information of the token.
			Example:
			"roles" : [{
			"name" : "role1",
			"id" : "roleid1"
			}, {
			"name" : "role2",
			"id" : "roleid2"
			}
			]

- Example response

The following is a sample request for obtaining a token for **user A**. The login password of the user is ********** and the domain name is **domain A**. The scope of the token is **domain**.

```
Token information stored in the response header:
X-Subject-Token:MIIDkgYJKoZIhvcNAQcCoIIDgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...

Token information stored in the response body:
{
  "token" : {
    "methods" : ["password"],
    "expires_at" : "2015-11-09T01:42:57.527363Z",
    "issued_at" : "2015-11-09T00:42:57.527404Z",
    "user" : {
      "domain" : {
      "id" : "ded485def148s4e7d2se41d5se...",
      "name" : "domain A"
      },
      "id" : "ee4dfb6e5540447cb37419051...",
      "name" : "user A",
      "password_expires_at":"2016-11-06T15:32:17.000000",
    },
    "domain" : {
       "name" : "domain A",
       "id" : "dod4ed5e8d4e8d2e8e8d5d2d..."
    },
    "catalog": [{
        "type": "identity",
        "id": "1331e5cff2a74d76b03da12259...",
        "name": "iam",
        "endpoints": [{
            "url": "https://sample.domain.com/v3",
            "region": "*",
            "region_id": "*",
           "interface": "public",
            "id": "089d4a381d574308a703122d3a..."
       }]
    }],
    "roles" : [{
       "name" : "role1",
       "id" : "roleid1"
       }, {
       "name" : "role2",
       "id" : "roleid2"
       }
  ]
  }
}
```

The following is a sample request for obtaining a token when virtual MFA–based login authentication is enabled.

```
Token information stored in the response header:
X-Subject-Token:MIIDkgYJKoZIhvcNAQcCoIIDgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...

Token information stored in the response body:
{
  "token": {
    "expires_at": "2020-09-05T06:50:44.390000Z",
    "mfa_authn_at": "2020-09-04T06:50:44.390000Z",
    "issued_at": "2020-09-04T06:50:44.390000Z",
     "methods": [
      "password",
      "totp"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "id": "33e1cbdd86d34e89a63cf8ad16a5f...",
```



```
            "interface": "public",
            "region": "*",
            "region_id": "*",
            "url": "https://sample.domain.com/v3.0"
          }
        ],
        "id": "100a6a3477f1495286579b819d399...",
        "name": "iam",
        "type": "iam"
      },
      ],
    "domain": {
      "id": "e6505630658e49649784759cdf251...",
      "name": "domain A"
    },
    "roles": [
     {
     "name" : "role1",
     "id" : "roleid1"
      },{
     "name" : "role1",
     "id" : "roleid1"

}
    ],
       "user": {
      "domain": {
        "id": "e6505630658e49649784759cdf251...",
        "name": "domain A"
      },
      "id": "092ac6365a0025b11f76c01e90100...",
      "name": "user A",
      "password_expires_at": ""
    }
  }
}
```


Status Codes

Status Code	Description
201	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error. The format may be incorrect.
503	Service unavailable.

4.2 Project Management

4.2.1 Querying Project Information Based on the Specified Criteria

Function

This API is used to query project information based on the specified criteria.

URI

- URI format

  GET /v3/projects{?domain_id,name,enabled,parent_id,is_domain,page,per_page}

- URI parameters

Parameter	Mandatory	Type	Description
domain_id	No	String	ID of an enterprise account to which a user belongs.
name	No	String	Project name.
parent_id	No	String	Parent project ID of a project.
enabled	No	Boolean	Whether a project is available.
is_domain	No	Boolean	Indicates whether the user calling the API is a tenant.
page	No	Integer	The page to be queried. The minimum value is **1**.
per_page	No	Integer	Number of data records on each page. Value range: [1,5000]

NOTE

When querying required information by page, ensure that the query parameters **page** and **per_page** both exist.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Fill **application/json;charset=utf8** in this field.
X-Auth-Token	Yes	String	Authenticated token of the target tenant.

- Example request
  *curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -X "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/projects?domain_id=5c9f5525d9d24c5bbf91e74d86772029&name=*region_name

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
projects	Yes	List	List of projects.
links	Yes	Object	Project resource link.

- Description for the project format

Parameter	Mandatory	Type	Description
is_domain	Yes	Boolean	Indicates whether the user calling the API is a tenant.
description	Yes	String	Project description.
links	Yes	Object	Project resource link.
enabled	Yes	Boolean	Whether a project is available.
id	Yes	String	Project ID.
parent_id	Yes	String	Parent ID of the project.
domain_id	Yes	String	ID of an enterprise account to which a project belongs.
name	Yes	String	Project name.

- Example response

```
{
  "links": {
    "self": "https://sample.domain.com/v3/projects?domain_id=c9f5525d9d24c5bbf91e74d86772029&name=region_name",
    "previous": null,
    "next": null
  },
  "projects": [
    {
      "is_domain": false,
      "description": "",
      "links": {
        "self": "https://sample.domain.com/v3/projects/e86737682ab64b2490c48f08bcc41914"
      },
      "enabled": true,
      "id": "e86737682ab64b2490c48f08bcc41914",
      "parent_id": "c9f5525d9d24c5bbf91e74d86772029",
      "domain_id": "c9f5525d9d24c5bbf91e74d86772029",
      "name": "region_name"
    }
  ]
}
```


Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.
503	Service unavailable.

5 Permissions Policies and Supported Actions

5.1 Introduction

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

An account has all the permissions required to call all APIs, but users must be assigned the required permissions. The permissions required for calling an API are determined by the actions supported by the API. Only users who have been granted permissions allowing the actions can call the API successfully. For example, if a user queries ECSs using an API, the user must have been granted permissions that allow the **ecs:servers:list** action.

Supported Actions

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

- Permission: Defined by actions in a custom policy.
- APIs: REST APIs that can be called in a custom policy.
- Actions: Added to a custom policy to control permissions for specific operations.
- IAM or enterprise projects: A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions supporting both IAM and enterprise projects can be assigned to user groups and take effect in both IAM and Enterprise Management. Policies that only contain actions supporting IAM projects can be assigned to user groups and only take effect for IAM. Such policies will not take effect if they are assigned to user groups in Enterprise Management. For details about the differences between IAM and enterprise projects, see "Differences Between IAM Projects and Enterprise Projects".

NOTE

- The check mark (√) and cross symbol (x) indicate that an action takes effect or does not take effect for the corresponding type of projects. A hyphen (-) indicates that an action is irrelevant to the corresponding type of projects.
- IAM is a global service which does not involve project-based authorization.
- Some permissions support only actions and do not support APIs.

6 Appendix

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

Status Code	Error Code	Error Message	Description	Measure
400	1100	Mandatory parameters are not specified.	Mandatory parameters are not specified.	Check the request parameters.
400	1101	Invalid username.	Invalid username.	Check the username.
400	1102	Invalid email address.	Invalid email address.	Check the email address.
400	1103	Incorrect password.	Incorrect password.	Check the password.
400	1104	Invalid mobile number.	Invalid mobile number.	Check the mobile number.
400	1105	The value of **xuser_type** must be the same as that of **xdomain_type**.	The value of **xuser_type** must be the same as that of **xdomain_type**.	Check whether the value of **xuser_type** is the same as that of **xdomain_type**.
400	1106	The country code and mobile number must be set at the same time.	The country code and mobile number must be set at the same time.	Check whether the country code and mobile number have been both specified.
400	1107	The account administrator cannot be deleted.	The account administrator cannot be deleted.	This operation is not allowed.
400	1108	The new password must be different from the old password.	The new password must be different from the old password.	Enter another password.
400	1109	The username already exists.	The username already exists.	Modify the username.

Status Code	Error Code	Error Message	Description	Measure
400	1110	The email address has already been used.	The email address has already been used.	Enter another email address.
400	1111	The mobile number has already been used.	The mobile number has already been used.	Enter another mobile number.
400	1113	The values of **xuser_id** and **xuser_type** already exist.	The values of **xuser_id** and **xuser_type** already exist.	Modify the values of **xuser_id** and **xuser_type**.
400	1115	The number of IAM users has reached the maximum allowed limit.	The number of IAM users has reached the maximum allowed limit.	Modify the user quota or contact technical support.
400	1117	Invalid user description.	Invalid user description.	Modify the user description.
400	1118	The password is weak.	The password is weak.	Enter another password.
400	IAM.0007	Request parameter %(key)s is invalid.	The request parameter is invalid.	Check the request parameter.
400	IAM.0008	Please scan the QR code first.	Scan the QR code first.	Scan the QR code first.
400	IAM.0009	X-Subject-Token is invalid in the request.	**X-Subject-Token** in the request is invalid.	Check the request parameter.
400	IAM.0010	The QR code has already been scanned by another user.	The QR code has already been scanned by someone else.	No action is required.
400	IAM.0011	Request body is invalid.	The request body is invalid.	Check the request body.

Status Code	Error Code	Error Message	Description	Measure
400	IAM.0072	'%(key)s' is a required property.	The request is invalid. For example, %(key)s is required.	Contact technical support.
400	IAM.0073	Invalid input for field '%(key)s'. The value is '%(value)s'.	The input is invalid.	Contact technical support.
400	IAM.0077	Invalid policy type.	The policy type is invalid.	Contact technical support.
400	IAM.1000	The role must be a JSONObject.	The role object is missing.	Check whether the request body contains the role object.
400	IAM.1001	The display_name must be a string and cannot be left blank or contain spaces.	The value of **display_name** is empty or contains spaces.	Check whether the value of **display_name** is correct.
400	IAM.1002	The length [input length] of the display name exceeds 64 characters.	The **display_name** field cannot exceed 64 characters.	Check the length of the **display_name** field.
400	IAM.1003	The display_name contains invalid characters.	The **display_name** field contains invalid characters.	Check whether the value of **display_name** is correct.
400	IAM.1004	The type must be a string and cannot be left blank or contain spaces.	The **type** field is empty.	Check whether the value of **type** is correct.

Status Code	Error Code	Error Message	Description	Measure
400	IAM.1005	Invalid type [input type].	The **type** field is invalid.	Check whether the value of **type** is correct.
400	IAM.1006	The custom policy does not need a catalog.	Custom policies cannot contain the **catalog** field.	Delete the **catalog** field.
400	IAM.1007	The custom policy does not need a flag.	Custom policies cannot contain the **flag** field.	Delete the **flag** field.
400	IAM.1008	The custom policy does not need a name.	Custom policies cannot contain the **name** field.	Delete the **name** field.
400	IAM.1009	The type of a custom policy must be 'AX' or 'XA'.	The type of a custom policy can only be **AX** or **XA**.	Change the value of the **type** field to **AX** or **XA**.
400	IAM.1010	The catalog must be a string.	The value of the **catalog** field must be a character string.	Check whether the value of **catalog** is correct.
400	IAM.1011	The length [input length] of the catalog exceeds 64 characters.	The **catalog** field cannot exceed 64 characters.	Check the length of the **catalog** field.
400	IAM.1012	Invalid catalog.	The **catalog** field is invalid.	Check whether the value of **catalog** is correct.
400	IAM.1013	The flag must be a string.	The value of the **flag** field must be a character string.	Check whether the value of **flag** is correct.

Status Code	Error Code	Error Message	Description	Measure
400	IAM.1014	The value of the flag must be 'fine_grained'.	The value of **flag** is not **fine_grained**.	Change the value of **flag** to **fine_grained**.
400	IAM.1015	The name must be a string and cannot be left blank or contain spaces.	The **name** field is empty.	Specify the **name** field for system-defined roles.
400	IAM.1016	The length of the name [input name] cannot exceed 64 characters.	The value of **name** cannot exceed 64 characters.	Check whether the value of **name** is correct.
400	IAM.1017	Invalid name.	The **name** field is invalid.	Check whether the value of **name** is correct.
400	IAM.1018	Invalid description.	The **description** field is invalid.	Check whether the value of **description** is correct.
400	IAM.1019	Invalid description_cn.	The **description_cn** field is invalid.	Check whether the value of **description_cn** is correct.
400	IAM.1020	The policy must be a JSONObject.	The **policy** object is missing.	Check whether the request body contains the **policy** object.
400	IAM.1021	The size [input policySize] of the policy exceeds 6,144 characters.	The **policy** object contains more than 6144 characters.	Check the length of the **policy** object.
400	IAM.1022	The length [input id length] of the ID exceeds 128 characters.	The **id** field contains more than 128 characters.	Check the length of the **id** field.

Status Code	Error Code	Error Message	Description	Measure
400	IAM.1023	Invalid ID '[input id]'.	The **id** field of the policy is invalid.	Check whether the value of **id** is correct.
400	IAM.1024	The version of a fine-grained policy must be '1.1'.	The version of the fine-grained policy is not 1.1.	Change the value of **version** to **1.1**.
400	IAM.1025	Fine-grained policies do not need depends.	The fine-grained policy contains the **depends** field.	Delete the **depends** field.
400	IAM.1026	The version of an RBAC policy must be '1.0' or '1.1'.	The version of an RBAC policy can only be 1.0 or 1.1.	Change the value of **version** to **1.0** or **1.1**.
400	IAM.1027	The Statement/ Rules must be a JSONArray.	The **statement** field is not a JSON array.	Check whether a JSON array statement exists.
400	IAM.1028	The number of statements [input statement size] must be greater than 0 and less than or equal to 8.	The policy does not contain any statements or contains more than 8 statements.	Ensure that the policy contains 1 to 8 statements.
400	IAM.1029	The value of Effect must be 'allow' or 'deny'.	The value of **effect** can only be **allow** or **deny**.	Set the **effect** field to **allow** or **deny**.
400	IAM.1030	The Action or NotAction must be a JSONArray.	The **action** or **notAction** field is invalid.	Check whether the value of **action** is correct.
400	IAM.1031	The Action and NotAction cannot be set at the same time in a statement.	The **action** and **notAction** fields cannot exist at the same time.	Delete the **action** or **notAction** field.

Status Code	Error Code	Error Message	Description	Measure
400	IAM.1032	The OCP NotAction cannot be 'allow'.	The **notAction** field cannot be **allow** for organization control policies (OCPs).	Specify the **notAction** field as **deny** for OCP policies.
400	IAM.1033	The number of actions [input action size] exceeds 100.	The number of actions exceeds 100.	Ensure that the number of actions does not exceed 100.
400	IAM.1034	The length [input urn length] of an action URN exceeds 128 characters.	An action contains more than 128 characters.	Ensure that each action does not exceed 128 characters.
400	IAM.1035	Action URN '[input urn]' contains invalid characters.	The action contains invalid characters.	Check whether the value of **action** is correct.
400	IAM.1036	Action '[input action]' has not been registered.	The action has not been registered.	Register the action using APIs of the registration center.
400	IAM.1037	The number of resource URIs [input Resource uri size] must be greater than 0 and less than or equal to 20.	Only 1 to 20 resources are allowed.	Check the number of resources.
400	IAM.1038	Resource URI '[input resource uri]' is invalid. Old resources only support agencies.	The resource URI is invalid.	Check whether each resource URI is correct.

Status Code	Error Code	Error Message	Description	Measure
400	IAM.1039	Old policies do not support conditions.	Old policies cannot contain the **condition** field.	Delete the **condition** field or use the new policy format.
400	IAM.1040	The number of resources [input Resource size] must be greater than 0 and less than or equal to 10.	Only 1 to 10 resource URIs are allowed.	Check the number of URIs of each resource object.
400	IAM.1041	The resource URI cannot be left blank or contain spaces.	A resource URI is empty.	Check whether each resource URI is correct.
400	IAM.1042	The length [input uri length] of a resource URI exceeds 1,500 characters.	A resource URI contains more than 1,500 characters.	Check the length of each resource URI.
400	IAM.1043	A region must be specified.	A region must be specified.	Specify a region in the resource URI.
400	IAM.1044	Region '[input resource region]' of resource '[input resource]' is invalid.	The **region** field is invalid.	Check whether the value of **region** is correct.
400	IAM.1045	Resource URI '[input resource uri]' or service '[input resource split]' is invalid.	The service name in the resource URI is invalid.	Check whether the service name is correct or register the service first.

Status Code	Error Code	Error Message	Description	Measure
400	IAM.1046	Resource URI '[input resource]' or resource type '[input resource split]' is invalid.	The resource type in the resource URI is invalid.	Check whether the resource type is correct or register the resource type first.
400	IAM.1047	Resource URI '[input resource uri]' contains invalid characters.	The resource URI is invalid.	Check whether the resource URI is correct.
400	IAM.1048	Resource URI '[input resource uri]' is too long or contains invalid characters.	The resource URI contains invalid characters.	Check whether the **id** field contains invalid characters.
400	IAM.1049	The Resource must be a JSONObject or JSONArray.	The **resource** object is missing.	Check whether the **resource** object is a JSON array.
400	IAM.1050	The number of conditions [input condition size] must be greater than 0 and less than or equal to 10.	Only 1 to 10 conditions are allowed.	Specify at least one condition or delete unused conditions.
400	IAM.1051	The values of Operator '[input operator]' cannot be null.	No operator is specified.	Enter a correct operator.
400	IAM.1052	Invalid Attribute '[input attribute]'.	The attribute is invalid.	Check the attribute value.

Status Code	Error Code	Error Message	Description	Measure
400	IAM.1053	Attribute '[input attribute]' must be a JSONArray.	The attribute is not a JSON array.	Check whether the attribute object is a JSON array.
400	IAM.1054	The number [input attribute size] of attributes '[input attribute]' for operator '[input operator]' must be greater than 0 and less than or equal to 10.	Each operator can only be used together with 1 to 10 attributes.	Check whether the number of attributes for each operator is correct.
400	IAM.1055	Attribute '[input attribute]' does not match operator '[input operator]'.	The attribute does not match the operator.	Check whether the attribute and operator match.
400	IAM.1056	The length [condition length] of attribute '[input attribute]' for operator '[input operator]' must be greater than 0 and less than or equal to 1024 characters.	Each condition can contain only 1 to 1024 characters.	Check the total length of the **condition** object.

Status Code	Error Code	Error Message	Description	Measure
400	IAM.1057	Value [input condition] of attribute [input attributes] for operator [input operator] contains invalid characters.	The **condition** field contains invalid characters.	Check whether the **condition** field contains invalid characters.
400	IAM.1058	The number of depends [input policyDepends size] exceeds 20.	The number of dependent permissions exceeds 20.	Delete excessive dependent permissions.
400	IAM.1059	Invalid key '{}'.	The policy contains an invalid key.	Modify or delete the invalid key in the policy request body.
400	IAM.1060	The value of key '{}' must be a string.	The value of this field must be a character string.	Change the values of **display_name** and **name** to character strings.
400	IAM.1061	Invalid TOTP passcode.	The authentication key is invalid.	Check the request or contact technical support.
400	IAM.1062	Login protection has been bound to mfa, the unbinding operation cannot be performed.	Login protection has been enabled and requires virtual MFA device based verification. You cannot unbind the virtual MFA device.	Check the request or contact technical support.
400	IAM.1101	The request body size %s is invalid.	The size of the request body does not meet the requirements.	Check whether the request body is empty or larger than 32 KB.

Status Code	Error Code	Error Message	Description	Measure
400	IAM.1102	The %s in the request body is invalid.	The value in the request body is incorrect.	Check the attribute value in the request body by referring to the *API Reference*.
400	IAM.1103	The %s is required in the request body.	The parameter is required but not specified in the request body.	Check the request body by referring to the *API Reference*.
400	IAM.1104	The access key %s is in the blacklist.	The AK in the request has been blacklisted.	Check whether the AK exists.
400	IAM.1105	The access key %s has expired.	The AK in the request has expired.	Create a new access key.
400	IAM.1106	The user %s with access key %s cannot be found.	The AK does not have matching user information.	Check whether the user or agency corresponding to the AK exists.
400	IAM.1107	The access key %s is inactive.	The AK in the request has been disabled.	Enable the AK.
400	IAM.1108	The securitytoken has expired.	The temporary access key has expired.	Obtain a new temporary access key.
400	IAM.1109	The project information cannot be found.	No project information can be found.	Check whether the project specified in the request body or token exists. If the fault persists, contact technical support.
401	IAM.0001	The request you have made requires authentication.	Authentication failed.	Complete or check the authentication information.

Status Code	Error Code	Error Message	Description	Measure
401	IAM.0061	Account locked.	The user has been locked.	Wait until the user is unlocked.
401	IAM.0062	Incorrect password.	Incorrect password.	Enter the correct password.
401	IAM.0063	Access token authenticatio n failed.	Access token authenticatio n failed.	Contact technical support.
401	IAM.0064	The access token does not have permissions for the request.	The IAM user does not have the required permissions.	Check the permissions of the IAM user.
401	IAM.0066	The token has expired.	The token has expired.	Use a valid token.
401	IAM.0067	Invalid token.	Invalid token.	Enter a valid token.
403	IAM.0002	You are not authorized to perform the requested action.	You do not have permission to perform this action.	Check whether you have been granted the permissions required to perform this action.
403	IAM.0003	Policy doesn't allow %(actions)s to be performed.	The action is not allowed in the policy.	Check whether the action is allowed in the policy.
403	IAM.0080	The user %s with access key %s is disabled.	The user corresponding to the AK has been disabled.	Contact the security administrator of the user.
403	IAM.0081	This user only supports console access, not programmatic access.	The user only has access to the management console.	Contact the security administrator of the user to change the user's access type.
403	IAM.0082	The user %s is disabled.	The user is disabled.	Contact the security administrator of the user.

Status Code	Error Code	Error Message	Description	Measure
403	IAM.0083	You do not have permission to access the private region %s.	You do not have permission to access private regions.	Select another region or contact the private region administrator.
404	IAM.0004	Could not find % (target)s: % (target_id)s.	The requested resource cannot be found.	Check the request or contact technical support.
409	IAM.0005	Conflict occurred when attempting to store % (type)s - % (details)s.	A conflict occurs when the requested resource is saved.	Check the request or contact technical support.
410	IAM.0020	Original auth failover to other regions, please auth downgrade	The Auth service in the original region is faulty and has switched to another region.	The system will automatically downgrade the authentication. No action is required.
429	IAM.0012	The throttling threshold has been reached. Threshold: %d times per %d seconds	The throttling threshold has been reached.	Check the request or contact technical support.
500	IAM.0006	An unexpected error prevented the server from fulfilling your request.	A system error occurred.	Contact technical support.

6.3 Obtaining User, Account, User Group, Project, and Agency Information

Obtaining User, Account, and Project Information

Your username, user ID, account name, account ID, project name, and project ID need to be specified in the URL and request body for calling certain APIs. Obtain these parameters on the **My Credentials** page.

Step 1 Log in to management console.

Step 2 Click the username in the upper right corner, and choose **My Credentials**.

Step 3 On the **My Credentials** page, view the username, user ID, account name, account ID, project name, and project ID.

----End

Obtaining User Group Information

Step 1 Log in to the IAM console, and choose **User Groups** in the navigation pane.

Step 2 Expand the details page of a user group and view the group name and ID.

----End

Obtaining Agency Information

Step 1 Log in to the IAM console, and choose **Agencies** in the navigation pane.

Step 2 Hover the mouse pointer over the agency you want to view. The name and ID of this agency are displayed.

----End

A Change History

Table A-1 Change history

Released On	Change History
2024-04-15	This issue is the first official release.

