

Identity and Access Management

API Reference

Issue 01
Date 2022-08-15



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.

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

Contents

1 Before You Start.....	1
1.1 Overview.....	1
1.2 API Calling.....	1
1.3 Endpoints.....	1
1.4 Constraints.....	2
1.5 Concepts.....	3
2 API Overview.....	5
3 Calling APIs.....	17
3.1 Making an API Request.....	17
3.2 Authentication.....	20
3.3 Response.....	21
4 APIs.....	24
4.1 Token Management.....	24
4.1.1 Obtaining a User Token.....	24
4.1.2 Obtaining an Agency Token.....	32
4.1.3 Verifying a Token.....	37
4.2 Access Key Management.....	42
4.2.1 Obtaining a Temporary AK/SK.....	42
4.2.2 Creating a Permanent Access Key.....	46
4.2.3 Listing Permanent Access Keys.....	49
4.2.4 Querying a Permanent Access Key.....	51
4.2.5 Modifying a Permanent Access Key.....	54
4.2.6 Deleting a Permanent Access Key.....	56
4.3 Region Management.....	58
4.3.1 Querying a Region List.....	58
4.3.2 Querying Region Details.....	60
4.4 Project Management.....	62
4.4.1 Querying Project Information Based on the Specified Criteria.....	62
4.4.2 Querying a User Project List.....	64
4.4.3 Querying the List of Projects Accessible to Users.....	67
4.4.4 Creating a Project.....	69
4.4.5 Modifying Project Data.....	70

4.4.6 Querying Information About a Specified Project.....	72
4.4.7 Setting the Status of a Specified Project.....	73
4.4.8 Querying Information and Status of a Specified Project.....	75
4.4.9 Querying the Quotas of a Project.....	77
4.5 Tenant Management.....	79
4.5.1 Querying the List of Domains Accessible to Users.....	79
4.5.2 Querying the Password Strength Policy.....	81
4.5.3 Querying the Password Strength Policy by Option.....	83
4.5.4 Querying a Resource Quota.....	85
4.6 User Management.....	86
4.6.1 Querying a User List.....	86
4.6.2 Querying User Details.....	90
4.6.3 Querying User Details (Recommended).....	93
4.6.4 Querying the User Group to Which a User Belongs.....	96
4.6.5 Querying Users in a User Group.....	98
4.6.6 Creating a User.....	102
4.6.7 Changing a Password.....	105
4.6.8 Modifying User Information.....	107
4.6.9 Modifying User Information (Including Email Address and Mobile Number).....	110
4.6.10 Modifying User Information (Including Email Address and Mobile Number).....	112
4.6.11 Deleting a User.....	118
4.6.12 Deleting a User from a User Group.....	119
4.6.13 Querying MFA Device Information of Users.....	121
4.6.14 Querying the MFA Device Information of a User.....	123
4.6.15 Querying Login Protection Configurations of Users.....	125
4.6.16 Querying the Login Protection Configuration of a User.....	127
4.6.17 Creating a Virtual MFA Device.....	130
4.6.18 Deleting a Virtual MFA Device.....	132
4.6.19 Binding a Virtual MFA Device.....	133
4.6.20 Unbinding a Virtual MFA Device.....	135
4.6.21 Modifying the Login Protection Configuration of a User.....	136
4.7 User Group Management.....	138
4.7.1 Listing User Groups.....	139
4.7.2 Querying User Group Details.....	140
4.7.3 Creating a User Group.....	142
4.7.4 Adding a User to a User Group.....	144
4.7.5 Updating a User Group.....	145
4.7.6 Deleting a User Group.....	147
4.7.7 Querying Whether a User Belongs to a User Group.....	148
4.8 Permission Management.....	149
4.8.1 Querying a Role List.....	149
4.8.2 Querying Role Details.....	154

4.8.3 Querying Permissions of a User Group Under a Domain.....	157
4.8.4 Querying Permissions of a User Group Corresponding to a Project.....	160
4.8.5 Granting Permissions to a User Group of a Domain.....	162
4.8.6 Granting Permissions to a User Group Corresponding to a Project.....	164
4.8.7 Deleting Permissions of a User Group Corresponding to a Project.....	165
4.8.8 Deleting Permissions of a User Group of a Domain.....	166
4.8.9 Querying Whether a User Group Under a Domain Has Specific Permissions.....	167
4.8.10 Querying Whether a User Group Corresponding to a Project Has Specific Permissions.....	168
4.8.11 Granting Permissions to a User Group for All Projects.....	169
4.8.12 Removing Specified Permissions of a User Group in All Projects.....	171
4.8.13 Checking Whether a User Group Has Specified Permissions for All Projects.....	172
4.8.14 Querying All Permissions of a User Group.....	173
4.9 Custom Policy Management.....	179
4.9.1 Listing Custom Policies.....	179
4.9.2 Querying Custom Policy Details.....	184
4.9.3 Creating a Custom Policy for Cloud Services.....	189
4.9.4 Creating a Custom Policy for Agencies.....	196
4.9.5 Modifying a Custom Policy for Cloud Services.....	202
4.9.6 Modifying a Custom Policy for Agencies.....	210
4.9.7 Deleting a Custom Policy.....	216
4.10 Agency Management.....	217
4.10.1 Creating an Agency.....	217
4.10.2 Querying an Agency List Based on the Specified Conditions.....	220
4.10.3 Obtaining Details of a Specified Agency.....	222
4.10.4 Modifying an Agency.....	224
4.10.5 Deleting an Agency.....	227
4.10.6 Granting Permissions to an Agency for a Project.....	228
4.10.7 Checking Whether an Agency Has the Specified Permissions on a Project.....	229
4.10.8 Querying the List of Permissions of an Agency on a Project.....	231
4.10.9 Deleting Permissions of an Agency on a Project.....	233
4.10.10 Granting Permissions to an Agency on a Domain.....	234
4.10.11 Checking Whether an Agency Has the Specified Permissions on a Domain.....	236
4.10.12 Querying the List of Permissions of an Agency on a Domain.....	237
4.10.13 Deleting Permissions of an Agency on a Domain.....	239
4.10.14 Querying All Permissions of an Agency.....	241
4.10.15 Granting Specified Permissions to an Agency for All Projects.....	243
4.10.16 Checking Whether an Agency Has Specified Permissions.....	244
4.10.17 Removing Specified Permissions of an Agency in All Projects.....	246
4.11 Security Settings.....	247
4.11.1 Querying the Operation Protection Policy.....	247
4.11.2 Modifying the Operation Protection Policy.....	249
4.11.3 Querying the Password Policy.....	252

4.11.4 Modifying the Password Policy.....	254
4.11.5 Querying the Login Authentication Policy.....	258
4.11.6 Modifying the Login Authentication Policy.....	261
4.11.7 Querying the ACL for Console Access.....	264
4.11.8 Modifying the ACL for Console Access.....	267
4.11.9 Querying the ACL for API Access.....	271
4.11.10 Modifying the ACL for API Access.....	274
4.12 Federated Identity Authentication Management.....	278
4.12.1 Obtaining a Token in Federated Identity Authentication Mode.....	278
4.12.1.1 SP Initiated.....	278
4.12.1.2 IdP Initiated.....	281
4.12.2 Identity Provider.....	287
4.12.2.1 Querying the Identity Provider List.....	287
4.12.2.2 Querying an Identity Provider.....	289
4.12.2.3 Creating a SAML Identity Provider.....	291
4.12.2.4 Updating a SAML Identity Provider.....	293
4.12.2.5 Deleting an Identity Provider.....	296
4.12.3 Mapping.....	297
4.12.3.1 Querying the Mapping List.....	297
4.12.3.2 Querying a Mapping.....	301
4.12.3.3 Creating a Mapping.....	305
4.12.3.4 Updating a Mapping.....	310
4.12.3.5 Deleting a Mapping.....	316
4.12.4 Protocol.....	317
4.12.4.1 Querying the Protocol List.....	317
4.12.4.2 Querying a Protocol.....	319
4.12.4.3 Registering a Protocol.....	321
4.12.4.4 Updating a Protocol.....	323
4.12.4.5 Deleting a Protocol.....	324
4.12.5 Metadata.....	326
4.12.5.1 Querying a Metadata File.....	326
4.12.5.2 Querying the Metadata File of Keystone.....	328
4.12.5.3 Importing a Metadata File.....	329
4.12.6 Token.....	330
4.12.6.1 Obtaining an Unscoped Token (SP Initiated).....	331
4.12.6.2 Obtaining an Unscoped Token (IdP Initiated).....	333
4.12.6.3 Obtaining a Scoped Token.....	335
4.12.7 Domain.....	343
4.12.7.1 Querying the List of Domains Accessible to Federated Users.....	343
4.12.8 Project.....	345
4.12.8.1 Querying the List of Projects Accessible to Federated Users.....	345
4.13 Version Information Management.....	346

4.13.1 Querying Keystone API Version Information.....	346
4.13.2 Querying Information About Keystone API Version 3.0.....	348
4.14 Services and Endpoints.....	349
4.14.1 Querying Services.....	349
4.14.2 Querying Service Details.....	352
4.14.3 Querying Endpoints.....	353
4.14.4 Querying Endpoint Details.....	355
4.14.5 Querying the Service Catalog.....	357
5 Permissions Policies and Supported Actions.....	359
5.1 Introduction.....	359
5.2 Action List.....	360
6 Appendix.....	371
6.1 Status Codes.....	371
6.2 Error Codes.....	375
6.3 Obtaining User, Account, User Group, Project, and Agency Information.....	389
A Change History.....	390

1 Before You Start

- [1.1 Overview](#)
- [1.2 API Calling](#)
- [1.3 Endpoints](#)
- [1.4 Constraints](#)
- [1.5 Concepts](#)

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 [3 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

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	4.1.1 Obtaining a User Token 4.1.2 Obtaining an Agency Token 4.12.6.3 Obtaining a Scoped Token
	GET /v3/auth/tokens	4.1.3 Verifying a Token
Access Key Management	POST /v3.0/OS-CREDENTIAL/securitytokens	4.2.1 Obtaining a Temporary AK/SK
Services and Endpoints	GET /v3/services{?type}	4.14.1 Querying Services
	GET /v3/endpoints{? interface, service_id}	4.14.3 Querying Endpoints
Version Information Management	GET /	4.13.1 Querying Keystone API Version Information
	GET /v3	4.13.2 Querying Information About Keystone API Version 3.0
Project Management	GET /v3/auth/projects	4.4.3 Querying the List of Projects Accessible to Users

Category	API URI	Link
Tenant Management	GET /v3/auth/domains	4.5.1 Querying the List of Domains Accessible to Users
Federated Identity Authentication Management	GET /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}/auth	4.12.6.1 Obtaining an Unscoped Token (SP Initiated)
	POST /v3.0/OS-FEDERATION/tokens	4.12.1.2 IdP Initiated
	GET /v3/OS-FEDERATION/projects	4.12.8.1 Querying the List of Projects Accessible to Federated Users
	GET /v3/OS-FEDERATION/domains	4.12.7.1 Querying the List of Domains Accessible to Federated Users
	GET /v3-ext/auth/OS-FEDERATION/SSO/metadata	4.12.5.2 Querying the Metadata File of Keystone

1.5 Concepts

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

- Account**

An account is created upon successful registration with the cloud platform. 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.
- 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**

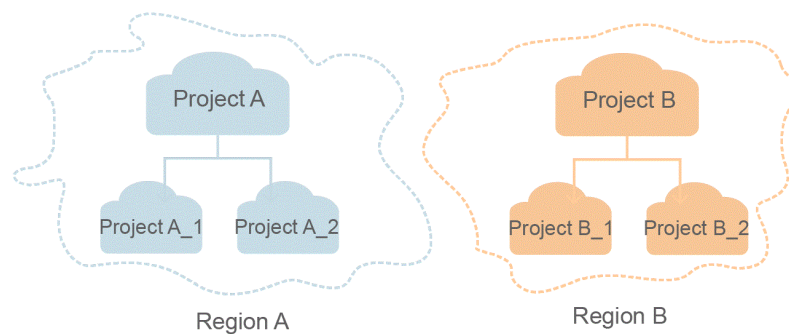
An AZ is a physical location where resources use independent power supplies and networks. A region contains one or more AZs that are physically isolated

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
4.1.1 Obtaining a User Token	Obtain a user token through username/password-based authentication.
4.1.2 Obtaining an Agency Token	Obtain an agency token.
Verifying a Token	Check the validity of a specified token. If the token is valid, detailed information about the token will be returned.
Verifying a Token	Provided for the administrator to verify the token of a user or provided for a user to verify their token. The administrator can only verify the token of a user created using the account. If the verified token is valid, 200 is displayed.
Deleting a User Token	Delete a token no matter whether the token has expired or not.

Access Key Management

API	Description
4.2.1 Obtaining a Temporary AK/SK	Obtain a temporary access key (AK/SK) and security token.
4.2.2 Creating a Permanent Access Key	Provided for the administrator to create a permanent access key for a user or provided for a user to create a permanent access key for themselves.
4.2.3 Listing Permanent Access Keys	Provided for the administrator to list all permanent access key of a user or provided for a user to list all of their permanent access keys.

API	Description
4.2.4 Querying a Permanent Access Key	Provided for the administrator to query the specified permanent access key of a user or provided for a user to query one of their permanent access keys.
4.2.5 Modifying a Permanent Access Key	Provided for the administrator to modify the specified permanent access key of a user or provided for a user to modify one of their permanent access keys.
4.2.6 Deleting a Permanent Access Key	Provided for the administrator to delete the specified permanent access key of a user or provided for a user to delete one of their permanent access keys.

Region Management

API	Description
4.3.1 Querying a Region List	List all regions.
4.3.2 Querying Region Details	Query region details.

Project Management

API	Description
4.4.1 Querying Project Information Based on the Specified Criteria	Query project information.
4.4.2 Querying a User Project List	Query the project list of a specified user.
4.4.3 Querying the List of Projects Accessible to Users	List the projects in which resources are accessible to a specified user.
4.4.4 Creating a Project	Create a project.
4.4.5 Modifying Project Data	Modify the details of a project.
4.4.6 Querying Information About a Specified Project	Query the detailed information about a project based on the project ID.

API	Description
4.4.7 Setting the Status of a Specified Project	Change the status of a specified project. The project status can be normal or suspended.
4.4.8 Querying Information and Status of a Specified Project	Query the details and status of a project.
4.4.9 Querying the Quotas of a Project	Query the quotas of a specified project.

Tenant Management

API	Description
4.5.1 Querying the List of Domains Accessible to Users	Query the list of domains accessible to users.
4.5.2 Querying the Password Strength Policy	Query the password strength policy, including its regular expression and description.
4.5.3 Querying the Password Strength Policy by Option	Query the regular expression or description of the password strength policy configured for a specified account.
4.5.4 Querying a Resource Quota	Query a resource quota. You can query the quota of users, user groups, identity providers, agencies, and policies.

User Management

API	Description
4.6.1 Querying a User List	List all users.
4.6.2 Querying User Details	Query the detailed information about a specified user.
4.6.3 Querying User Details (Recommended)	Provided for the administrator to query the details about a specified user or provided for a user to query their details.

API	Description
4.6.4 Querying the User Group to Which a User Belongs	Query the information about the groups to which a specified user belongs.
4.6.5 Querying Users in a User Group	Provided for the administrator to query the users in a user group.
4.6.6 Creating a User	Create a user under a domain.
4.6.7 Changing a Password	Change the password for a user.
4.6.8 Modifying User Information	Modify user information under a domain.
4.6.9 Modifying User Information (Including Email Address and Mobile Number)	Provided for users to modify their information.
4.6.10 Modifying User Information (Including Email Address and Mobile Number)	Provided for the administrator to modify user information.
4.6.11 Deleting a User	Provided for the administrator to delete a user.
4.6.12 Deleting a User from a User Group	Delete a user from a user group.
4.6.13 Querying MFA Device Information of Users	Provided for the administrator to query the MFA device information of users.
4.6.14 Querying the MFA Device Information of a User	Provided for the administrator to query the MFA device information of a specified user or provided for a user to query their MFA device information.
4.6.15 Querying Login Protection Configurations of Users	Provided for the administrator to query the login protection configurations of users.

API	Description
4.6.16 Querying the Login Protection Configuration of a User	Provided for the administrator to query the login protection configuration of a specified user or provided for a user to query their login protection configuration.
4.6.19 Binding a Virtual MFA Device	Bind a virtual MFA device to a user.
4.6.20 Unbinding a Virtual MFA Device	Unbind the virtual MFA device bound to a user.
4.6.17 Creating a Virtual MFA Device	Create a virtual MFA device for a user.
4.6.18 Deleting a Virtual MFA Device	Provided for the administrator to delete the virtual MFA device created for a user.
4.6.21 Modifying the Login Protection Configuration of a User	Provided for the administrator to modify the login protection configuration of a user.

User Group Management

API	Description
4.7.1 Listing User Groups	Provided for the administrator to list all user groups.
4.7.2 Querying User Group Details	Provided for the administrator to query user group information.
4.7.3 Creating a User Group	Provided for the administrator to create a user group.
4.7.4 Adding a User to a User Group	Provided for the administrator to add a user to a specified user group.
4.7.5 Updating a User Group	Provided for the administrator to update user group information.
4.7.6 Deleting a User Group	Provided for the administrator to delete a user group.
4.7.7 Querying Whether a User Belongs to a User Group	Provided for the administrator to check whether a user belongs to a specified user group.

Permission Management

API	Description
4.8.1 Querying a Role List	Provided for the administrator to list all permissions.
4.8.2 Querying Role Details	Provided for the administrator to query permission information.
4.8.3 Querying Permissions of a User Group Under a Domain	Query the permissions of a specified user group under a domain.
4.8.4 Querying Permissions of a User Group Corresponding to a Project	Query the permissions of a specified user group for a project.
4.8.5 Granting Permissions to a User Group of a Domain	Grant permissions to a specified user group under a domain.
4.8.6 Granting Permissions to a User Group Corresponding to a Project	Grant permissions to a specified user group for a project.
4.8.7 Deleting Permissions of a User Group Corresponding to a Project	Delete permissions of a user group corresponding to a project.
4.8.8 Deleting Permissions of a User Group of a Domain	Delete permissions of a specified user group of a domain.
4.8.9 Querying Whether a User Group Under a Domain Has Specific Permissions	Query whether a specified user group under a domain has specific permissions.
4.8.10 Querying Whether a User Group Corresponding to a Project Has Specific Permissions	Query whether a user group corresponding to a project has specific permissions.

API	Description
4.8.11 Granting Permissions to a User Group for All Projects	Grant permissions to a user group.
4.8.12 Removing Specified Permissions of a User Group in All Projects	Provided for the administrator to remove the specified permissions of a user group in all projects.
4.8.13 Checking Whether a User Group Has Specified Permissions for All Projects	Provided for the administrator to check whether a user group has specified permissions for all projects.
4.8.14 Querying All Permissions of a User Group	Provided for the administrator to query all permissions that have been assigned to a user group.

Custom Policy Management

API	Description
4.9.1 Listing Custom Policies	Provided for the administrator to list all custom policies.
4.9.2 Querying Custom Policy Details	Provided for the administrator to query custom policy details.
4.9.3 Creating a Custom Policy for Cloud Services	Provided for the administrator to create a custom policy for cloud services.
4.9.4 Creating a Custom Policy for Agencies	Provided for the administrator to create a custom policy for agencies.
4.9.5 Modifying a Custom Policy for Cloud Services	Provided for the administrator to modify a custom policy for cloud services.
4.9.6 Modifying a Custom Policy for Agencies	Provided for the administrator to modify a custom policy for agencies.
4.9.7 Deleting a Custom Policy	Provided for the administrator to delete a custom policy.

Agency Management

API	Description
4.10.1 Creating an Agency	Create an agency.
4.10.2 Querying an Agency List Based on the Specified Conditions	Query an agency list based on the specified conditions.
4.10.3 Obtaining Details of a Specified Agency	Query the details of a specified agency.
4.10.4 Modifying an Agency	Modify agency information, including the trust_domain_id , description , and trust_domain_name parameters.
4.10.5 Deleting an Agency	Delete an agency.
4.10.6 Granting Permissions to an Agency for a Project	Grant permissions to an agency for a project.
4.10.7 Checking Whether an Agency Has the Specified Permissions on a Project	Check whether an agency has the specified permissions on a project.
4.10.8 Querying the List of Permissions of an Agency on a Project	Query the list of permissions of an agency on a project.
4.10.9 Deleting Permissions of an Agency on a Project	Delete permissions of an agency on a project.
4.10.10 Granting Permissions to an Agency on a Domain	Grant permissions to an agency on a domain.
4.10.11 Checking Whether an Agency Has the Specified Permissions on a Domain	Check whether an agency has the specified permissions on a domain.
4.10.12 Querying the List of Permissions of an Agency on a Domain	Query the list of permissions of an agency on a domain.

API	Description
4.10.13 Deleting Permissions of an Agency on a Domain	Delete permissions of an agency on a domain.
4.10.14 Querying All Permissions of an Agency	Provided for the administrator to query all permissions that have been assigned to an agency.
4.10.15 Granting Specified Permissions to an Agency for All Projects	Provided for the administrator to grant specified permissions to an agency for all projects.
4.10.16 Checking Whether an Agency Has Specified Permissions	Provided for the administrator to check whether an agency has specified permissions.
4.10.17 Removing Specified Permissions of an Agency in All Projects	Provided for the administrator to remove the specified permissions of an agency in all projects.

Security Settings

API	Description
4.11.1 Querying the Operation Protection Policy	Query the operation protection policy.
4.11.2 Modifying the Operation Protection Policy	Provided for the administrator to modify the operation protection policy.
4.11.3 Querying the Password Policy	Query the password policy.
4.11.4 Modifying the Password Policy	Provided for the administrator to modify the password policy.
4.11.5 Querying the Login Authentication Policy	Query the login authentication policy.
4.11.6 Modifying the Login Authentication Policy	Provided for the administrator to modify the login authentication policy.
4.11.7 Querying the ACL for Console Access	Query the ACL for console access.

API	Description
4.11.8 Modifying the ACL for Console Access	Provided for the administrator to modify the ACL for console access.
4.11.9 Querying the ACL for API Access	Query the ACL for API access.
4.11.10 Modifying the ACL for API Access	Provided for the administrator to modify the ACL for API access.

Federated Identity Authentication Management

API	Description
4.12.1.1 SP Initiated	Obtain a federated authentication token using the OpenStack Client or ShibbolethECP Client.
4.12.1.2 IdP Initiated	Obtain a federated authentication token in the IdP-initiated mode. The Client4ShibbolethIdP script is used as an example.
4.12.2.1 Querying the Identity Provider List	List all identity providers.
4.12.2.2 Querying an Identity Provider	Query the details about an identity provider.
4.12.2.3 Creating a SAML Identity Provider	Provided for the administrator to create an identity provider.
4.12.2.4 Updating a SAML Identity Provider	Provided for the administrator to update an identity provider.
4.12.2.5 Deleting an Identity Provider	Provided for the administrator to delete an identity provider.
4.12.3.1 Querying the Mapping List	List all mappings.
4.12.3.2 Querying a Mapping	Query the information about a mapping.
4.12.3.3 Creating a Mapping	Provided for the administrator to register a mapping.
4.12.3.4 Updating a Mapping	Provided for the administrator to update a mapping.
4.12.3.5 Deleting a Mapping	Provided for the administrator to delete a mapping.

API	Description
4.12.4.1 Querying the Protocol List	List all protocols.
4.12.4.2 Querying a Protocol	Query the details of a protocol.
4.12.4.3 Registering a Protocol	Provided for the administrator to register a protocol, that is, to associate a protocol with an identity provider.
4.12.4.4 Updating a Protocol	Provided for the administrator to update the protocol associated with a specified identity provider.
4.12.4.5 Deleting a Protocol	Provided for the administrator to delete the protocol associated with a specified identity provider.
4.12.5.1 Querying a Metadata File	Provided for the administrator to query the metadata file imported to IAM for an identity provider.
4.12.5.2 Querying the Metadata File of Keystone	Query the metadata file of Keystone.
4.12.5.3 Importing a Metadata File	Provided for the administrator to import a metadata file.
4.12.6.1 Obtaining an Unscoped Token (SP Initiated)	Obtain an unscoped token through SP-initiated federated identity authentication.
4.12.6.2 Obtaining an Unscoped Token (IdP Initiated)	Obtain an unscoped token through IdP-initiated federated identity authentication.
4.12.6.3 Obtaining a Scoped Token	Obtain a scoped token through federated identity authentication.
4.12.7.1 Querying the List of Domains Accessible to Federated Users	List the accounts whose resources are accessible to federated users.
4.12.8.1 Querying the List of Projects Accessible to Federated Users	List the projects in which resources are accessible to federated users.

Version Information Management

API	Description
4.13.1 Querying Keystone API Version Information	Query the version information of Keystone APIs.
4.13.2 Querying Information About Keystone API Version 3.0	Obtain the information about Keystone API 3.0.

Services and Endpoints

API	Description
4.14.1 Querying Services	List all services.
4.14.2 Querying Service Details	Query the details of a service.
4.14.5 Querying the Service Catalog	Query the service catalog corresponding to X-Auth-Token contained in the request.
4.14.3 Querying Endpoints	List all endpoints.
4.14.4 Querying Endpoint Details	Query the details of an endpoint.

3 Calling APIs

3.1 Making an API Request

3.2 Authentication

3.3 Response

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 [4.1.1 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

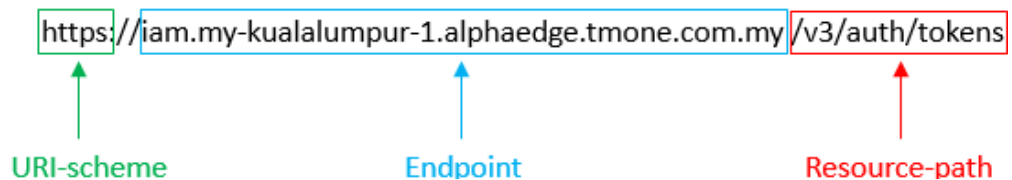
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from Regions and Endpoints. For example, the endpoint of IAM in the my-kualalumpur-1 region is iam.my-kualalumpur-1.alphaedge.tmone.com.my .
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 .

Parameter	Description
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 **my-kualalumpur-1** region, obtain the endpoint of IAM (`iam.my-kualalumpur-1.alphaedge.tmone.com.my`) for this region and the **resource-path** (`/v3/auth/tokens`) in the URI of the API used to obtain a user token (see [4.1.1 Obtaining a User Token](#)). Then, construct the URI as follows:

```
https://iam.my-kualalumpur-1.alphaedge.tmone.com.my/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 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 ([4.1.1 Obtaining a User Token](#)), the request method is POST. The request is as follows:

```
POST https://iam.my-kualalumpur-1.alphaedge.tmone.com.my/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 ([4.1.1 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 [3.2 Authentication](#).

The API used to obtain a user token ([4.1.1 Obtaining a User Token](#)) 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.my-kualalumpur-1.alphaedge.tmone.com.my/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 ([4.1.1 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*, *domiannname*, ******* (login password), and *xxxxxxxxxxxxxxxxxxxx* (project ID) with the actual values. If you obtain a token using an account, ensure that you set *username* and *domiannname* 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 [4.1.1 Obtaining a User Token](#).

```
POST https://iam.my-kualalumpur-1.alphaedge.tmone.com.my/v3/auth/tokens
Content-Type: application/json
{
```

```

"auth": {
  "identity": {
    "methods": [
      "password"
    ],
    "password": {
      "user": {
        "name": "username",
        "password": "*****",
        "domain": {
          "name": "domiannname"
        }
      }
    }
  },
  "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 ([4.1.1 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 [4.1.1 Obtaining a User Token](#). IAM APIs can be called only by using a global service token. To call the API described in [4.1.1 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://iam.my-kualalumpur-1.alphaedge.tmone.com.my/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. 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 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 [6.1 Status Codes](#).

For example, if status code **201** is returned for calling the API used to obtain a user token ([4.1.1 Obtaining a User Token](#)), the request is successful.

Response Header

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

[4.1.1 Obtaining a User Token](#) shows the response header fields for the API used to obtain a user token ([Figure 3-2](#)). 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
→ MIIYXQYJKoZIhvcNAQcCoIIYtJCCGEOCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOansiZXhwaXJlc19hdCI6IjIwMTk0MDtMTNUMMC
fj3Kjs6YgKnpVNRbW2eZ5eb78SZ0kqjACgkklqO1wi4JlGzrpd18LGXK5bdfq4lqHCYb8P4NaYONyEjcAgzIVeFYtLWT1GSO0zxKZmlQHq82HBqHdglZO9fuEbl5dMhdavj+33wEI
xHRC9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXI1jipPEGA270g1FruooL6jggIFkNPQuFSOU8+uSsttVwRtnfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUUVhVpxk8pxiX1wTEboX-
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 ([4.1.1 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](#)
- [4.2 Access Key Management](#)
- [4.3 Region Management](#)
- [4.4 Project Management](#)
- [4.5 Tenant Management](#)
- [4.6 User Management](#)
- [4.7 User Group Management](#)
- [4.8 Permission Management](#)
- [4.9 Custom Policy Management](#)
- [4.10 Agency Management](#)
- [4.11 Security Settings](#)
- [4.12 Federated Identity Authentication Management](#)
- [4.13 Version Information Management](#)
- [4.14 Services and Endpoints](#)

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**

The validity period of a token is **24 hours**. Cache the token to prevent frequent API calling. 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 to which the user belongs 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 . <pre>"identity": { "methods": ["password"], "password": {</pre>
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"] .

Parameter	Mandatory	Type	Description
password	Yes	JSON object	<p>Authentication information.</p> <p>Example:</p> <pre>"password": { "user": { "name": "user A", "password": "*****#", "domain": { "name": "domain A" } } }</pre> <ul style="list-style-type: none"> ● 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.
totp	No	JSON object	<p>Authentication information. This parameter is mandatory only when virtual MFA-based login authentication is enabled.</p> <p>Example:</p> <pre>"totp": { "user": { "id": "b95b78b67fa045b38104c12fb...", "passcode": "*****" } }</pre> <ul style="list-style-type: none"> ● 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.

Parameter	Mandator y	Type	Description
scope	No	JSON object	<p>Usage scope of the token. The value can be project or domain.</p> <ul style="list-style-type: none"> • 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. <pre>"scope": { "project": { "id": "0b95b78b67fa045b38104c12fb..." } }</pre> • 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. <pre>"scope": { "domain": { "name": " domain A" } }</pre>

- 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"
      }
    }
  }
}
```

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.
expires_at	Yes	String	Expiration date of the token.
issued_at	Yes	String	Time when the token was issued.
user	Yes	JSON object	<p>Example:</p> <pre>"user": { "name": "user A", "id": "b95b78b67fa045b38104...", "password_expires_at": "2016-11-06T15:32:17.000000", "domain": { "name": "domain A", "id": "fdec73ffea524aa1b373e40..." } }</pre> <ul style="list-style-type: none"> 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	<p>This parameter is returned only when the scope parameter in the request body has been set to domain.</p> <p>Example:</p> <pre>"domain": { "name": "domain A" "id": "fdec73ffea524aa1b373e40..."</pre> <ul style="list-style-type: none"> • domain.name: Name of the domain that created the user. • domain.id: ID of the domain.
project	No	JSON object	<p>This parameter is returned only when the scope parameter in the request body has been set to project.</p> <p>Example:</p> <pre>"project": { "name": "project A", "id": "34c77f3eaf84c00aaf54...", "domain": { "name": "domain A", "id": "fdec73ffea524aa1b373e40..." } }</pre> <ul style="list-style-type: none"> • 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	<p>Endpoint information.</p> <p>Example:</p> <pre>"catalog": [{ "type": "identity", "id": "1331e5cff2a74d76b03da1225910e...", "name": "iam", "endpoints": [{ "url": "https:// sample.domain.com/v3", "region": "*", "region_id": "*", "interface": "public", "id": "089d4a381d574308a703122d3ae73..." } }]</pre> <ul style="list-style-type: none"> ● type: Type of the service to which the API belongs. ● 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	<p>Permissions information of the token.</p> <p>Example:</p> <pre>"roles": [{ "name": "role1", "id": "roleid1" }, { "name": "role2", "id": "roleid2" }]</pre>

- 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"
  }
]
}
```

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.

Status Code	Description
503	Service unavailable.

4.1.2 Obtaining an Agency Token

Function

This API is used to obtain an agency token. For example, after a trust relationship is established between A and B, A is the delegating party and B is the delegated party. Then B can use this API to obtain the agency token. The agency token can be used to manage only the resources that B is delegated to manage. To manage their resources, B needs to obtain a user token according to [4.1.1 Obtaining a User Token](#).

NOTE

The validity period of a token is **24 hours**. Cache the token to prevent frequent API calling. 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.

URI

POST /v3/auth/tokens

Request Parameters

- Parameters in the request header

Parameter	Mandator y	Type	Description
Content-Type	Yes	String	Fill application/json; charset=utf8 in this field.
X-Auth-Token	Yes	String	Token that assigns the permissions of the Agent Operator policy to user B.

- Parameters in the request body

Parameter	Mandator y	Type	Description
identity	Yes	JSON object	Authentication parameters, including: methods and assume_role . <pre>"identity": { "methods": ["assume_role"], "assume_role": {</pre>

Parameter	Mandator y	Type	Description
methods	Yes	String Array	Method for obtaining the token. Set this field to assume_role .
domain_name or domain_id	Yes	String	Domain name or domain ID of the delegating party A. Specify either domain_name or domain_id .
xrole_name	Yes	String	Name of the agency created by A.
scope	No	JSON object	Usage scope of the token. The value can be project or domain . <ul style="list-style-type: none"> If this field is set to project, the token can only be used to access resources in the project of a specified ID or name. <pre> "scope": { "project": { "id": "0b95b78b67fa045b38104c12fb..." } } </pre> If this field is set to domain, the token can be used to access all resources under the domain of a specified ID or name. <pre> "scope": { "domain": { "id": "6b8eb224c76842e3ac2..." } } </pre>

- Example request

The following is a sample request for obtaining an agency token for **domain A**. The name of the agency is **agencytest**.

```

{
  "auth":{
    "identity":{
      "methods":[
        "assume_role"
      ],
      "assume_role":{
        "domain_name":"domain A",
        "xrole_name":"agencytest"
      }
    }
  },
  "scope":{
    "domain":{
      "name":"domain A"
    }
  }
}

```



```
}
}
```

Response Parameters

- Parameters in the response header

Parameter	Mandatory	Type	Description
X-Subject-Token	Yes	String	Agency token that is obtained.

- Token format description

Parameter	Mandatory	Type	Description
methods	Yes	Json Array	Method for obtaining the token.
expires_at	Yes	String	Expiration date of the token.
issued_at	Yes	String	Time when the token was issued.
user	Yes	JSON object	<p>Detailed information about the delegating party. Example:</p> <pre>"user": { "name": "user A", "id": "userid", "password_expires_at": "2016-11-06T15:32:17.000000", "domain": { "name": "domain A", "id": "domainid" } }</pre> <ul style="list-style-type: none"> user.name: Username of the delegating party. user.id: User ID of the delegating party. domain.name: Name of the domain to which the delegating party belongs. domain.id: ID of the domain. password_expires_at: Time when the password will expire. null indicates that the password will not expire. This parameter is optional.

Parameter	Mandatory	Type	Description
domain	No	JSON object	<p>This parameter is returned only when the scope parameter in the request body has been set to domain.</p> <p>Example:</p> <pre>"domain": { "name": "domain A", "id": "domainid" }</pre> <ul style="list-style-type: none"> • domain.name: Name of the domain to which the delegating party belongs. • domain.id: ID of the domain.
project	No	JSON object	<p>This parameter is returned only when the scope parameter in the request body has been set to project.</p> <p>Example:</p> <pre>"project": { "name": "projectname", "id": "projectid" }</pre> <ul style="list-style-type: none"> • project.name: Name of a project. • project.id: ID of the project.
catalog	No	Json Array	<p>Endpoint information.</p> <p>Example:</p> <pre>"catalog": [{ "type": "identity", "id": "1331e5cff2a74d76b03da1225910e31d", "name": "iam", "endpoints": [{ "url": "https://sample.domain.com/ v3", "region": "*", "region_id": "*", "interface": "public", "id": "089d4a381d574308a703122d3ae738e9" }] }]</pre>

Parameter	Mandator y	Type	Description
roles	Yes	JSON object	Permissions information of the token. Example: <pre>"roles" : [{ "name" : "role1", "id" : "roleid1" }, { "name" : "role2", "id" : "roleid2" }]</pre>
assumed_by	Yes	JSON object	Detailed information about the delegated party. Example: Example: <pre>"assumed_by": { "user": { "domain": { "name": "domain B", "id": "bfdd55e02a014894b5a2693f31..." }, "name": "user B", "id": "ff5ea657f1dd45c4b8f398cab..." } }</pre> <ul style="list-style-type: none"> • domain.name: Name of the domain to which the delegated party belongs. • user.name: Username of the delegated party.

- **Example response**

Token information stored in the response header:
X-Subject-Token:MIIDkgYJKoZIhvcNAQcCoIIDgzCCA38CAQExDTALBglhkgBZQMEAgEwgXXXXX...

X-Frame-Options: SAMEORIGIN

Information included in the response body:

```
{
  "token": {
    "methods": [
      "assume_role"
    ],
    "issued_at": "2017-05-18T11:44:05.232000Z",
    "expires_at": "2017-05-19T11:44:05.232000Z",
    "user": {
      "id": "93e12ecdad6f4abd84968741da...",
      "name": "user A/agencytest",
      "password_expires_at": "2016-11-06T15:32:17.000000",
      "domain": {
        "id": "ce925c42c25943bebbba10ea64a...",
        "name": "domain A"
      }
    }
  },
  "domain": {
    "id": "ce925c42c25943bebbba10ea64a...",
    "name": "domain A"
  }
}
```

```

},
"roles": [
  {
    "id": "c11c61319f08404eaf94f8030b9...",
    "name": "role1"
  },
  {
    "id": "d52dde35ijg62fex2ijhdc785sc3...",
    "name": "role2"
  },
  {
    "id": "d862dwd32dwhu854rdcs447ed1d7..."
    "name": "op_gated_tasssg6"
  }
],
"assumed_by": {
  "user": {
    "domain": {
      "name": "domain B",
      "id": "c1a78a82d81c4a19b03bfe82d3ad..."
    },
    "id": "cdeb158dda854cc3bab77d8926ff...",
    "name": "User B"
  }
}
}
}

```

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.
503	Service unavailable.

4.1.3 Verifying a Token

Function

This API is used to check the validity of a specified token. If the token is valid, detailed information about the token will be returned.

URI

GET /v3/auth/tokens

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	<ul style="list-style-type: none"> To verify your own token, specify your token. There are no special requirements on the permissions that your token must have. To verify the token of another user under the same domain, use a token that has permissions of the Security Administrator policy.
X-Subject-Token	Yes	String	Token to be verified.

- Query parameters

Parameter	Mandatory	Type	Description
nocatalog	No	String	If this parameter is set, no catalog information will be displayed in the response.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H "X-Subject-Token:$token" -X GET https://sample.domain.com/v3/auth/tokens
```

Response Parameters

- Parameters in the response header

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

- Parameters in the response body

Parameter	Mandatory	Type	Description
token	Yes	Object	Token information list.

- Token format description

Parameter	Mandatory	Type	Description
methods	Yes	Array	Method of obtaining the token, for example, password .
expires_at	Yes	String	Expiration date of the token.
issued_at	Yes	String	Time when the token was issued.
user	Yes	Object	<p>Example:</p> <pre>"user": { "name": "username", "id": "userid", "password_expires_at": "2016-11-06T15:32:17.000000", "domain": { "name": "domainname", "id": "domainid" } }</pre> <ul style="list-style-type: none"> • user.name: Name of the user that owns the token. • user.id: ID of the user. • domain.name: Name of the domain to which the user belongs. • domain.id: ID of the domain. • password_expires_at: Time when the password will expire. null indicates that the password will not expire. This parameter is optional.
domain	No	Object	<p>The system determines whether to return this field based on the scope contained in the request for obtaining the token.</p> <p>Example:</p> <pre>"domain": { "name": "domainname", "id": "domainid" }</pre> <ul style="list-style-type: none"> • domain.name: Domain name. • domain.id: Domain ID.

Parameter	Mandatory	Type	Description
project	No	Object	<p>The system determines whether to return this field based on the scope contained in the request for obtaining the token.</p> <p>Example:</p> <pre>"project": { "name": "projectname", "id": "projectid", }</pre> <ul style="list-style-type: none"> ● project.name: Name of a project. ● project.id: ID of the project.
catalog	No	Json Array	<p>Endpoint information.</p> <p>Example:</p> <pre>"catalog": [{ "type": "identity", "id": "1331e5cff2a74d76b03da1225910e31d", "name": "iam", "endpoints": [{ "url": "https://sample.domain.com/ v3", "region": "*", "region_id": "*", "interface": "public", "id": "089d4a381d574308a703122d3ae738e9" }] }]</pre> <ul style="list-style-type: none"> ● type: Type of the service to which the API belongs. ● 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.

Parameter	Mandator y	Type	Description
roles	Yes	Array	Permissions information of the token. Example: <pre>"roles": [{ "name": "role1", "id": "roleid1" }, { "name": "role2", "id": "roleid2" }]</pre>

- Example response

```
{
  "token": {
    "methods": ["password"],
    "expires_at": "2015-11-09T01:42:57.527363Z",
    "issued_at": "2015-11-09T00:42:57.527404Z",
    "user": {
      "domain": {
        "id": "default",
        "name": "Default"
      },
      "id": "ee4dfb6e5540447cb3741905149XXX...",
      "password_expires_at": "2016-11-06T15:32:17.000000",
      "name": "admin"
    },
    "domain": {
      "name": "Default",
      "id": "default"
    },
    "roles": [{
      "name": "role1",
      "id": "roleid1"
    }, {
      "name": "role2",
      "id": "roleid2"
    }
  ]
}
```

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.
503	Service unavailable.

4.2 Access Key Management

4.2.1 Obtaining a Temporary AK/SK

Function

You can obtain a temporary AK/SK and security token (offline AK/SK) by using a user token, agency token, and federated token. A temporary AK/SK is a token with temporary permissions issued to users. It conforms to the principle of least privilege and can be used to temporarily access OBS.

URI

POST /v3.0/OS-CREDENTIAL/securitytokens

Request Parameters

- Parameters in the request header
 - Obtaining a temporary AK/SK with an agency token (**methods** is set to **assume_role**)

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with permissions of the Agent Operator policy.
Content-Type	Yes	String	Fill application/json;charset=utf8 in this field.

- Obtaining a temporary AK/SK with a user token or a federated token (**methods** is set to **token**)

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Fill application/json;charset=utf8 in this field.
X-Auth-Token	No	String	User token or federated token required for obtaining a temporary AK/SK. You need to specify either this parameter or the token ID in the request body. This parameter takes the precedence.

- Parameters in the request body
 - Obtaining a temporary AK/SK with an agency token (**methods** is set to **assume_role**)

Parameter	Mandatory	Type	Description
methods	Yes	String Array	Fill assume_role in this field.
agency_name	Yes	String	Name of the agency created by a delegating party.
domain_name or domain_id	Yes	String	domain.name: Name of the domain to which the delegating party belongs.
duration_seconds	No	Int	Validity period (in seconds) of an AK/SK and security token. The value ranges from 15 minutes to 24 hours. The default value is 15 minutes.

Parameter	Mandatory	Type	Description
scope	No	Object	<p>AK/SK and security token. If this parameter is left blank, the generated security token does not contain the scope information. You are advised to leave this parameter blank. To set the scope of the temporary AK/SK and security token, specify a project or domain.</p> <ul style="list-style-type: none"> If this field is set to project, the temporary AK/SK and security token can only be used to access resources in the project of a specified ID or name. <pre> "scope": { "project": { "id": "0b95b78b67fa045b38104c12f b..." } } </pre> If this field is set to domain, the temporary AK/SK and security token can be used to access all resources under the domain of a specified ID or name. <pre> "scope": { "domain": { "name": " domain A" } } </pre>

- Obtaining a temporary AK/SK with a user token or a federated token (**methods** is set to **token**)

Parameter	Mandatory	Type	Description
methods	Yes	String Array	Fill token in this field.

Parameter	Mandatory	Type	Description
token	No	JSON object	Common token or federated token required for obtaining a temporary AK/SK. You need to choose either the ID in this object or X-Auth-Token in the request header. X-Auth-Token takes priority over the ID in this object.
duration_se conds	No	Int	Validity period (in seconds) of an AK/SK and security token. The value ranges from 15 minutes to 24 hours. The default value is 15 minutes.

- Example request
 - When the **methods** parameter is set to **assume_role**

```
{
  "auth": {
    "identity": {
      "methods": [
        "assume_role"
      ],
      "assume_role": {
        "domain_id": "411edb4b634144f587ffc88f9bbdxxx",
        "xrole_name": "testagency",
        "duration_seconds": "3600"
      }
    }
  }
}
```

- When the **methods** parameter is set to **token**

```
{
  "auth": {
    "identity": {
      "methods": [
        "token"
      ],
      "token": {
        "id":
"MIIDKgYJKoZIhvcNAQcCoIIDgzCCA38CAQExDTALBgIghkgBZQMEAgEwgXXXXX...",
        "duration_seconds": "900"
      }
    }
  }
}
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
credential	Yes	Object	Authentication information.

- Description about the credential content.

Parameter	Mandatory	Type	Description
expires_at	Yes	String	Expiration time.
access	Yes	String	AK.
secret	Yes	String	SK.
securitytoken	Yes	String	Used for subsequent replacement of an SK or token.

- Example response

```
{
  "credential": {
    "access": "NQC51NFINJS1JXX...",
    "secret": "EY74MByPZ46kTRJL9ay5DskqXX...",
    "expires_at": "2017-04-17T07:55:18.575000Z",
    "securitytoken": "gAAAAABY9GbWUaGtoa9DPj7_dE4qUSnAXXX..."
  }
}
```

Status Codes

Status Code	Description
201	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
500	The system is abnormal.

4.2.2 Creating a Permanent Access Key

Function

This API can be used by the administrator to create a permanent access key for an IAM user or used by an IAM user to create a permanent access key for itself.

Access keys are identity credentials for using development tools (APIs, CLI, and SDKs) to access the cloud system. Access keys cannot be used to log in to the

console. AK is used in conjunction with an SK to sign requests cryptographically, ensuring that the requests are secret, complete, and correct.

The API can be called using both the global endpoint and region-specific endpoints.

URI

POST /v3.0/OS-CREDENTIAL/credentials

Request Parameters

Table 4-1 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	A token with Security Administrator permissions is required if the administrator is requesting to create a permanent access key for an IAM user. The user token (no special permission requirements) of an IAM user is required if the user is requesting to create a permanent access key for itself.

Table 4-2 Parameters in the request body

Parameter	Mandatory	Type	Description
credential	Yes	Object	Authentication information.

Table 4-3 credential

Parameter	Mandatory	Type	Description
user_id	Yes	String	IAM user ID.
description	No	String	Description of the access key.

Response Parameters

Table 4-4 Parameters in the response body

Parameter	Type	Description
credential	Object	Authentication result.

Table 4-5 credential

Parameter	Type	Description
create_time	String	Time when the access key was created.
access	String	AK.
secret	String	SK.
status	String	Status of the access key.
user_id	String	IAM user ID.
description	String	Description of the access key.

Example Request

```
POST https://sample.domain.com/v3.0/OS-CREDENTIAL/credentials
{
  "credential": {
    "description": "IAMDescription",
    "user_id": "07609fb9358010e21f7bc003751c7c32"
  }
}
```

Example Response

Status code: 201

The request is successful.

```
{
  "credential": {
    "access": "P83EVBZJMXCYTMUI...",
    "create_time": "2020-01-08T06:25:19.014028Z",
    "user_id": "07609fb9358010e21f7bc003751...",
    "description": "IAMDescription",
    "secret": "TTqAHPbhWorg9ozx8Dv9MUyzYnOKDppxzHt...",
    "status": "active"
  }
}
```

Status code: 400

The server failed to process the request. (The number of access keys has reached the maximum allowed limit.)

```
{
  "error": {
```

```

    "message": "akSkNumExceed",
    "code": 400,
    "title": "Bad Request"
  }
}

```

Status Codes

Status Code	Description
201	The request is successful.
400	The server failed to process the request. (The number of access keys has reached the maximum allowed limit.)
401	Authentication failed.
403	Access denied.
500	Internal server error.

Error Codes

None

4.2.3 Listing Permanent Access Keys

Function

This API can be used by the administrator to list all permanent access key of an IAM user or used by an IAM user to list all of their permanent access keys.

The API can be called using both the global endpoint and region-specific endpoints.

URI

GET /v3.0/OS-CREDENTIAL/credentials

Table 4-6 Query parameters

Parameter	Mandatory	Type	Description
user_id	No	String	User ID.

Request Parameters

Table 4-7 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	A token with Security Administrator permissions is required if the administrator is requesting to query all permanent access keys of an IAM user. The user token (no special permission requirements) of an IAM user is required if the user is requesting to query their permanent access keys.

Response Parameters

Table 4-8 Parameters in the response body

Parameter	Type	Description
credentials	Array of objects	Authentication result.

Table 4-9 credentials

Parameter	Type	Description
user_id	String	IAM user ID.
access	String	AK.
status	String	Status of the access key.
create_time	String	Time when the access key was created.
description	String	Description of the access key.

Example Request

- Request for an IAM user to query their permanent access keys
GET https://sample.domain.com/v3.0/OS-CREDENTIAL/credentials
- Request for an administrator to query all permanent access keys of an IAM user (user ID: **07609fb9358010e21f7bc003751c...**)
GET https://sample.domain.com/v3.0/OS-CREDENTIAL/credentials?user_id=07609fb9358010e21f7bc0037....

Example Response

Status code: 200

The request is successful.

```
{
  "credentials": [
    {
      "access": "LOSZM4YRVLKOY9E8X...",
      "create_time": "2020-01-08T06:26:08.123059Z",
      "user_id": "07609fb9358010e21f7bc0037...",
      "description": "",
      "status": "active"
    },
    {
      "access": "P83EVBZJMXCYTMU...",
      "create_time": "2020-01-08T06:25:19.014028Z",
      "user_id": "07609fb9358010e21f7bc003751...",
      "description": "",
      "status": "active"
    }
  ]
}
```

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.

Error Codes

None

4.2.4 Querying a Permanent Access Key

Function

This API can be used by the administrator to query the specified permanent access key of an IAM user or used by an IAM user to query one of their permanent access keys.

The API can be called using both the global endpoint and region-specific endpoints.

URI

GET /v3.0/OS-CREDENTIAL/credentials/{access_key}

Table 4-10 URI parameters

Parameter	Mandatory	Type	Description
access_key	Yes	String	AK of the access key to be queried.

Request Parameters

Table 4-11 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	A token with Security Administrator permissions is required if the administrator is requesting to query a specified permanent access key of an IAM user. The user token (no special permission requirements) of an IAM user is required if the user is requesting to query one of their permanent access keys.

Response Parameters

Table 4-12 Parameters in the response body

Parameter	Type	Description
credential	Object	Authentication result.

Table 4-13 credential

Parameter	Type	Description
user_id	String	IAM user ID.
access	String	AK.

Parameter	Type	Description
status	String	Status of the access key.
create_time	String	Time when the access key was created.
last_use_time	String	Time when the access key was last used.
description	String	Description of the access key.

Example Request

```
GET https://sample.domain.com/v3.0/OS-CREDENTIAL/credentials/{access_key}
```

Example Response

Status code: 200

The request is successful.

```
{
  "credential": {
    "last_use_time": "2020-01-08T06:26:08.123059Z",
    "access": "LOSZM4YRVLKOY9E8...",
    "create_time": "2020-01-08T06:26:08.123059Z",
    "user_id": "07609fb9358010e21f7bc003751...",
    "description": "",
    "status": "active"
  }
}
```

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.

Error Codes

None

4.2.5 Modifying a Permanent Access Key

Function

This API can be used by the administrator to modify the specified permanent access key of an IAM user or used by an IAM user to modify one of their permanent access keys.

The API can be called using both the global endpoint and region-specific endpoints.

URI

PUT /v3.0/OS-CREDENTIAL/credentials/{access_key}

Table 4-14 URI parameters

Parameter	Mandatory	Type	Description
access_key	Yes	String	AK of the access key to be modified.

Request Parameters

Table 4-15 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	A token with Security Administrator permissions is required if the administrator is requesting to modify a specified permanent access key of an IAM user. The user token (no special permission requirements) of an IAM user is required if the user is requesting to modify one of their permanent access keys.

Table 4-16 Parameters in the request body

Parameter	Mandatory	Type	Description
credential	Yes	Object	Authentication information.

Table 4-17 credential

Parameter	Mandatory	Type	Description
status	No	String	Status of the access key to be changed to The value can be active or inactive . Options: <ul style="list-style-type: none"> • active • inactive
description	No	String	Description of the access key

Response Parameters

Table 4-18 Parameters in the response body

Parameter	Type	Description
credential	Object	Authentication information.

Table 4-19 credential

Parameter	Type	Description
user_id	String	IAM user ID.
access	String	AK.
status	String	Status of the access key.
create_time	String	Time when the access key was created.
description	String	Description of the access key.

Example Request

```
PUT https://sample.domain.com/v3.0/OS-CREDENTIAL/credentials/{access_key}
{
  "credential": {
```

```
"status": "inactive",  
"description": "IAMDescription"  
}  
}
```

Example Response

Status code: 200

The request is successful.

```
{  
  "credential": {  
    "status": "inactive",  
    "access": "LOSZM4YRVLKOY9...",  
    "create_time": "2020-01-08T06:26:08.123059Z",  
    "user_id": "07609fb9358010e21f7bc00375..."  
  }  
}
```

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.

Error Codes

None

4.2.6 Deleting a Permanent Access Key

Function

This API can be used by the administrator to delete the specified permanent access key of an IAM user or used by an IAM user to delete one of their permanent access keys.

The API can be called using both the global endpoint and region-specific endpoints.

URI

DELETE /v3.0/OS-CREDENTIAL/credentials/{access_key}

Table 4-20 URI parameters

Parameter	Mandatory	Type	Description
access_key	Yes	String	AK to be deleted.

Request Parameters

Table 4-21 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	A token with Security Administrator permissions is required if the administrator is requesting to delete a specified permanent access key of an IAM user. The user token (no special permission requirements) of an IAM user is required if the user is requesting to delete one of their permanent access keys.

Response Parameters

None

Example Request

DELETE https://sample.domain.com/v3.0/OS-CREDENTIAL/credentials/{access_key}

Example Response

None

Status Codes

Status Code	Description
204	The access key is deleted successfully.
400	The server failed to process the request.
401	Authentication failed.

Status Code	Description
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

Error Codes

None

4.3 Region Management

4.3.1 Querying a Region List

Function

This API is used to query a region list.

URI

GET /v3/regions

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. If the token does not contain the private region information, the system does not return the private region in the query result.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/regions
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
links	Yes	Dict	Region resource link.
regions	Yes	List	Region list.

- Description for the regions format

Parameter	Mandatory	Type	Description
description	Yes	String	Region description.
parent_region_id	Yes	String	Parent region ID of a region.
id	Yes	String	Region ID.
locales	Yes	Dict	Region name.
type	No	String	Region type.
links	Yes	Dict	Region resource link.

- Example response (successful response)

```
{
  "regions": [
    {
      "parent_region_id": null,
      "description": "",
      "links": {
        "self": "None/v3/regions/1500365963661574434"
      },
      "type": "private",
      "id": "1500365963661574434",
      "locales": {
        "en-us": "region_name2"
      }
    },
    {
      "parent_region_id": null,
      "description": "",
      "links": {
        "self": "https://sample.domain.com/v3/regions/500017826026667755"
      },
      "type": "private",
      "id": "500017826026667755",
      "locales": {
        "en-us": "region_name2"
      }
    },
    {
      "parent_region_id": null,
      "description": "",
      "links": {
        "self": "https://sample.domain.com/v3/regions/region_name"
      },
      "type": "public",
      "id": "test2",
      "locales": {

```

```

        "en-us": "region_name2"
    }
  },
  {
    "parent_region_id": null,
    "links": {
      "self": "https://sample.domain.com/v3/regions/test1112244"
    },
    "id": "test1112244",
    "locales": {
      "en-us": "testregion1"
    },
    "description": ""
  }
],
"links": {
  "self": "https://sample.domain.com/v3/regions",
  "previous": null,
  "next": null
}
}

```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.3.2 Querying Region Details

Function

This API is used to query region details.

URI

- URI format
GET /v3/regions/{region_id}

- URI parameters

Parameter	Mandatory	Type	Description
region_id	Yes	String	Region ID.

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.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H 'X-Auth-Token:$token' -X GET https://sample.domain.com/v3/regions/test-pusb999999991
```

Response Parameters

Example response

```
{
  "region": {
    "parent_region_id": null,
    "description": "",
    "links": {
      "self": "https://sample.domain.com/v3/regions/test-pusb999999991"
    },
    "type": "public",
    "id": "test-pusb999999991",
    "locales": {
      "en-us": "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.

Status Code	Description
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.4 Project Management

4.4.1 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.

4.4.2 Querying a User Project List

Function

This API is used to query the project list of a specified user.

URI

- URI format
GET /v3/users/{user_id}/projects
- URI parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.

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 with the Security Administrator permission or token of the user.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/users/43cbe5e77aaf4665bbb962062dc1fc9d/projects
```

Response Parameters

- Parameters in the response body

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

- Description for the project format

Parameter	Mandatory	Type	Description
description	Yes	String	Project description.
id	Yes	String	Project ID.
domain_id	Yes	String	ID of the domain where a project is located.
name	Yes	String	Project name.
links	Yes	Object	Project resource link.
is_domain	Yes	Boolean	Indicates whether the user calling the API is a tenant.
enabled	Yes	Boolean	Whether a project is available.

Parameter	Mandatory	Type	Description
parent_id	Yes	String	Parent ID of the project.

- Example response

```
{
  "links": {
    "self": "https://sample.domain.com/v3/auth/projects",
    "previous": null,
    "next": null
  },
  "projects": [
    {
      "is_domain": false,
      "description": "",
      "links": {
        "self": "https://sample.domain.com/v3/projects/9041929bcc6e4bfe85add4e7b96ffd7"
      },
      "enabled": true,
      "id": "9041929bcc6e4bfe85add4e7b96ffd7",
      "parent_id": "398998b5392f4150ad48fe456d6de4f1",
      "domain_id": "398998b5392f4150ad48fe456d6de4f1",
      "name": "region_name"
    },
    {
      "is_domain": false,
      "description": "",
      "links": {
        "self": "https://sample.domain.com/v3/projects/ee65ca70d3cf43aaa1ea6492ce15f289"
      },
      "enabled": true,
      "id": "ee65ca70d3cf43aaa1ea6492ce15f289",
      "parent_id": "398998b5392f4150ad48fe456d6de4f1",
      "domain_id": "398998b5392f4150ad48fe456d6de4f1",
      "name": "MOS" //Default project name of OBS
    }
  ]
}
```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.

Status Code	Description
500	Internal server error.
503	Service unavailable.

4.4.3 Querying the List of Projects Accessible to Users

Function

This API is used to query the list of projects accessible to users.

URI

GET /v3/auth/projects

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 user.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET /v3/auth/projects
```

Response Parameters

- Parameters in the response body

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

- Description for the project format

Parameter	Mandatory	Type	Description
description	Yes	String	Project description.
id	Yes	String	ID of a project.

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the domain where a project is located.
name	Yes	String	Project name.
links	Yes	Object	Project resource link.
is_domain	Yes	Boolean	Indicates whether the user calling the API is a tenant.
enabled	Yes	Boolean	Whether a project is available.
parent_id	Yes	String	Parent ID of the project.

- Example response

```
{
  "links": {
    "self": "https://sample.domain.com/v3/auth/projects",
    "previous": null,
    "next": null
  },
  "projects": [
    {
      "is_domain": false,
      "description": "",
      "links": {
        "self": "https://sample.domain.com/v3/projects/9041929bcc6e4bfe85add4e7b96ffdd7"
      },
      "enabled": true,
      "id": "9041929bcc6e4bfe85add4e7b96ffdd7",
      "parent_id": "398998b5392f4150ad48fe456d6de4f1",
      "domain_id": "398998b5392f4150ad48fe456d6de4f1",
      "name": "region"
    },
    {
      "is_domain": false,
      "description": "",
      "links": {
        "self": "https://sample.domain.com/v3/projects/ee65ca70d3cf43aaa1ea6492ce15f289"
      },
      "enabled": true,
      "id": "ee65ca70d3cf43aaa1ea6492ce15f289",
      "parent_id": "398998b5392f4150ad48fe456d6de4f1",
      "domain_id": "398998b5392f4150ad48fe456d6de4f1",
      "name": "{project_name}"
    }
  ]
}
```

Status Codes

Status Code	Description
200	The request is successful.

Status Code	Description
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.4.4 Creating a Project

Function

This API is used to create a project.

URI

POST /v3/projects

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
name	Yes	String	Project name, which must start with " <i>ID of an existing region_</i> " and be less than or equal to 64 characters. Example: <i>{region_id}_test1</i>

Parameter	Mandatory	Type	Description
parent_id	Yes	String	Parent project ID to which a project belongs.
domain_id	No	String	ID of the domain that a project belongs to.
description	No	String	Project description, which can contain a maximum of 255 characters.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X POST -d '{"project":{"domain_id":"acf2ffabba974fae8f30378ffde2c...","name":"region_test1"}}' https://sample.domain.com/v3/projects
```

Response Parameters

Example response

```
{
  "project": {
    "is_domain": false,
    "description": "",
    "links": {
      "self": "https://sample.domain.com/v3/projects/3de1461665f045ef91ba1efe8121b979"
    },
    "enabled": true,
    "id": "3de1461665f045ef91ba1efe8121b979",
    "parent_id": "d1294857fdf64251994892b344f53e88",
    "domain_id": "d1294857fdf64251994892b344f53e88",
    "name": "region_test1"
  }
}
```

Status Codes

Status Code	Description
201	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
409	Duplicate project name.

4.4.5 Modifying Project Data

Function

This API is used to modify project information.

URI

- URI format
PATCH /v3/projects/{project_id}
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Text type and encoding mode. Fill application/json; charset=utf8 in this field.
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
name	No	String	Project name, which must start with the ID of an existing region and be less than or equal to 64 characters. Example: <i>{region}_test2</i>
description	No	String	Project description, which can contain a maximum of 255 characters.

- Example Request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X PATCH -d '{"project":{"name":"region_test2","description":"test_project_desc"}}' https://sample.domain.com/v3/projects/23da5961c8214f5caf701c27d9703959
```

Response Parameters

Example Response

```
{
  "project": {
    "is_domain": false,
    "description": "test_project_desc",
    "links": {
      "self": "https://sample.domain.com/v3/projects/23da5961c8214f5caf701c27d9703959"
    }
  },
}
```

```

"enabled": true,
"id": "23da5961c8214f5caf701c27d9703959",
"parent_id": "d1294857fdf64251994892b344f53e88",
"domain_id": "d1294857fdf64251994892b344f53e88",
"name": "region_test2"
}
}

```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
409	Duplicate project name.

4.4.6 Querying Information About a Specified Project

Function

This API is used to query detailed information about a project based on the project ID.

URI

- URI format
GET /v3/projects/{project_id}
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

Request Parameters

- Parameters in the request header

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

- Example request

Obtaining information about the project whose ID is project_id=619d3e78f61b4be68bc5aa0b59edcf7b

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/projects/619d3e78f61b4be68bc5aa0b59edcf7b
```

Response Parameters

- Example response

```
{
  "project": {
    "is_domain": false,
    "description": "",
    "links": {
      "self": "https://sample.domain.com/v3/projects/2e93d63d8d2249f5a4ac5e2c78586a6e"
    },
    "enabled": true,
    "id": "2e93d63d8d2249f5a4ac5e2c78586a6e",
    "parent_id": "44c0781c83484eb9a4a5d4d233522cea",
    "domain_id": "44c0781c83484eb9a4a5d4d233522cea",
    "name": "MOS" //Default project name of OBS
  }
}
```

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.

4.4.7 Setting the Status of a Specified Project

Function

This API is used to set the status of a specified project. The project statuses include **Normal** and **Suspended**.

URI

- URI format
PUT /v3-ext/projects/{project_id}
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
status	Yes	String	Project status. The value can be suspended or normal .

NOTE

- suspended**: The project is frozen.
 - normal**: The project is normal or unfrozen.
- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -X "X-Auth-Token:$token" -X PUT -d '{"project":{"status":"suspended"}}'https://sample.domain.com/v3-ext/projects/5c9f5525d9d24c5bbf91e74d86772029
```

Response Parameters

None

Status Codes

Status Code	Description
204	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.

Status Code	Description
404	The requested resource cannot be found.
500	Internal server error.
503	Service unavailable.

4.4.8 Querying Information and Status of a Specified Project

Function

This API is used to query details about a specified project, including the project status.

URI

- URI format
GET /v3-ext/projects/{project_id}
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

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 with the Security Administrator permission.

- 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-ext/projects/5c9f5525d9d24c5bbf91e74d86772029
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
project	Yes	Object	Project information.

- Description for the project format

Parameter	Mandatory	Type	Description
description	Yes	String	Project description.
id	Yes	String	Project ID.
domain_id	Yes	String	ID of the domain that a project belongs to.
name	Yes	String	Project name.
is_domain	Yes	Boolean	Indicates whether the user calling the API is a tenant.
enabled	Yes	Boolean	Whether a project is available.
parent_id	Yes	String	Parent ID of a project.
status	Yes	String	Project status.
suspended_time	No	String	Time when a project is suspended.

- Example response

```
{
  "project": {
    "is_domain": false,
    "description": "",
    "enabled": true,
    "id": "ee65ca70d3cf43aaa1ea6492ce15f289",
    "parent_id": "9041929bcc6e4bfe85add4e7b96ffdd7",
    "domain_id": "398998b5392f4150ad48fe456d6de4f1",
    "name": "{region_id}_test1",
    "status": "suspended",
    "suspended_time": "2017-08-17T02:50:23.000000"
  }
}
```

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.

4.4.9 Querying the Quotas of a Project

Function

This API is used to query the quotas of a specified project.

URI

- URI format
GET /v3.0/OS-QUOTA/projects/{project_id}
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	ID of the project to query quotas.

Request Parameters

Table 4-22 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Provide either of the following tokens: <ul style="list-style-type: none"> • Token with Security Administrator permissions • IAM user token with the scope specified as the project you want to query

Response Parameters

Table 4-23 Parameters in the response body

Parameter	Type	Description
quotas	object	Quota information of the domain.

Table 4-24 quotas

Parameter	Type	Description
resources	Array of objects	Resource information.

Table 4-25 resources

Parameter	Type	Description
max	Integer	Maximum quota.
min	Integer	Minimum quota.
quota	Integer	Current quota.
type	String	Quota type.
used	Integer	Used quota.

Example Request

GET https://sample.domain.com/v3.0/OS-QUOTA/projects/{project_id}

Example Response

Status code: 200

The request is successful.

```
{
  "quotas": {
    "resources": [
      {
        "max": 50,
        "min": 0,
        "quota": 10,
        "type": "project",
        "used": 4
      }
    ]
  }
}
```

Status code: 403

Access denied.

- Example 1

```
{
  "error_msg": "You are not authorized to perform the requested action.",
  "error_code": "IAM.0002"
}
```

- Example 2

```
{
  "error_msg": "Policy doesn't allow %(actions)s to be performed.",
}
```

```
{
  "error_code" : "IAM.0003"
}
```

Status code: 404

The requested resource cannot be found.

```
{
  "error_msg" : "Could not find %(target)s: %(target_id)s.",
  "error_code" : "IAM.0004"
}
```

Status code: 500

Internal server error.

```
{
  "error_msg" : "An unexpected error prevented the server from fulfilling your request.",
  "error_code" : "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.5 Tenant Management

4.5.1 Querying the List of Domains Accessible to Users

Function

This API is used to query the list of domains accessible to users.

URI

GET /v3/auth/domains

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H 'X-Auth-Token:$token' -X GET https://sample.domain.com/v3/auth/domains
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
domains	Yes	JSONArray	List of domains.
links	Yes	JSON object	Domain resource link.

- Description for the domain format

Parameter	Mandatory	Type	Description
enabled	Yes	Boolean	Whether a domain is enabled. true indicates that the domain is enabled. false indicates that the domain is disabled. The default value is true .
id	Yes	String	Domain ID.
name	Yes	String	Domain name.
links	Yes	JSON object	Domain resource link.
description	No	String	Domain description.

- Example response

```
{
  "domains": [{
    "description": "desc of domain",
    "enabled": true,
    "id": "37ef61",
    "links": {
      "self": "https://sample.domain.com/v3/domains/37ef61"
    },
    "name": "my domain"
  }],
  "links": {
    "self": "https://sample.domain.com/v3/auth/domains",
    "previous": null,
    "next": null
  }
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.5.2 Querying the Password Strength Policy

Function

This API is used to query the password strength policy, including its regular expression and description.

URI

- URI format
GET /v3/domains/{domain_id}/config/security_compliance
- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID to be queried.

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 a user.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/domains/{domain_id}/config/security_compliance
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
security_compliance	Yes	JSON	Password strength policy.
password_regex	Yes	String	Regular expression of the password strength policy.
password_regex_description	Yes	String	Description of the password strength policy.

- Example response

```
{
  "config": {
    "security_compliance": {
      "password_regex": "^(?=.*\\d)(?=.*[a-zA-Z]).{7,}$",
      "password_regex_description": "Passwords must contain at least 1 letter, 1 digit, and be a minimum length of 7 characters."
    }
  }
}
```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.5.3 Querying the Password Strength Policy by Option

Function

This API is used to query the password strength policy by **option**. The option can be the regular expression and description of the password strength policy.

URI

- URI format
GET /v3/domains/{domain_id}/config/security_compliance/{option}
- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the domain whose password strength policy is to be queried.
option	Yes	String	Query option, which can be password_regex or password_regex_description .

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 user token.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/domains/{domain_id}/config/security_compliance/password_regex
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
config	Yes	JSON	Password strength policy of a domain.

- Description for the config format

Parameter	Mandatory	Type	Description
password_regex	No	String	Regular expression of the password strength policy (When option is set to password_regex).
password_regex_description	No	String	Description of the password strength policy (When option is set to password_regex_description).

- Example response

When **option** is set to **password_regex**:

```
{
  "config": {
    "password_regex": "^(?=.*\\d)(?=.*[a-zA-Z]).{7,}$"
  }
}
```

When **option** is set to **password_regex_description**:

```
{
  "config": {
    "password_regex_description": "Passwords must contain at least 1 letter, 1 digit, and be a minimum length of 7 characters."
  }
}
```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.5.4 Querying a Resource Quota

Function

This API is used to query a resource quota. You can query the quota of users, user groups, identity providers, agencies, and policies.

URI

- URI format
GET /v3.0/OS-QUOTA/domains/{domain_id}
- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the domain whose quota is to be queried.
type	No	String	Type of the quota to be queried. The value can be user, group, idp, agency, and policy.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token (no special permission requirements).

- Example request
GET https://sample.domain.com/v3.0/OS-QUOTA/domains/{domain_id}?type=group

Response Parameters

Table 4-26 Parameters in the response body

Parameter	Type	Description
quotas	Object	Quota information of the domain.

Table 4-27 quotas

Parameter	Type	Description
resources	Array of objects	Resource information.

Table 4-28 resources

Parameter	Type	Description
max	Integer	Maximum quota.
min	Integer	Minimum quota.
quota	Integer	Current quota.
type	String	Quota type.
used	Integer	Used quota.

- Example response

Group quota:

```
{
  "quotas": {
    "resources": [
      {
        "max": 200,
        "min": 10,
        "quota": 20,
        "type": "group",
        "used": 6
      }
    ]
  }
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
500	Internal server error.
503	Service unavailable.

4.6 User Management

4.6.1 Querying a User List

Function

This API is used to query a user list.

URI

- URI format
GET /v3/users
- URI parameters

Parameter	Mandatory	Type	Description
domain_id	No	String	ID of the domain that a user belongs to.
enabled	No	String	Whether a user is enabled. true indicates that the user is enabled. false indicates that the user is disabled. The default value is true .
name	No	String	Username.

Parameter	Mandatory	Type	Description
password_expires_at	No	String	<p>Password expiration time. The format is password_expires_at=operator.timestamp.</p> <p>Example: password_expires_at=lt:2016-12-08T22:02:00Z</p> <ul style="list-style-type: none"> The value of operator can be lt, lte, gt, gte, eq, or neq. <ul style="list-style-type: none"> lt: The expiration time is earlier than <i>timestamp</i>. lte: The expiration time is earlier than or equal to <i>timestamp</i>. gt: The expiration time is later than <i>timestamp</i>. gte: The expiration time is equal to or later than <i>timestamp</i>. eq: The expiration time is equal to <i>timestamp</i>. neq: The expiration time is not equal to <i>timestamp</i>. The timestamp format is YYYY-MM-DDTHH:mm:ssZ.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/users
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
users	Yes	JSONArray	User list.
links	Yes	JSON object	Links of a user resource.

- Description for the user format

Parameter	Mandatory	Type	Description
description	Yes	String	Description for a user.
domain_id	Yes	String	ID of the tenant that the user belongs to.
enabled	Yes	Boolean	Indicates whether the user is enabled. The value can be true or false . The default value is true .
id	Yes	String	User ID.
links	Yes	JSON object	User resource link.
name	Yes	String	Username.
password_expires_at	Yes	String	UTC time when the password will expire. null indicates that the password will not expire.
pwd_status	No	Boolean	Password status. true means that the password needs to be changed, and false means that the password is normal.
pwd_strength	No	String	Password strength. The value can be high , mid , or low .
default_project_id	No	String	ID of the project that is displayed by default when the user logs in to the console.
last_project_id	No	String	ID of the project that the user lastly accessed before exiting the system.
email	No	String	User email address.

- Example response

```
{
  "users": [{
    "name": "username",
    "links": {
      "self": "https://sample.domain.com/v3/users/6d8b04e3bf99445b8f76300xxx"
    },
    "description": "1234",
    "domain_id": "88b16b6440684467b8825d7xxx",
    "enabled": false,
    "id": "6d8b04e3bf99445b8f763009xxx",
    "email": "",
    "default_project_id": "263fd9",
    "password_expires_at": "2016-12-07T00:00:00.000000Z",
    "pwd_status": true,
    "pwd_strength": "high",
    "last_project_id": ""
  }],
  "links": {
    "self": "https://sample.domain.com/v3/users?domain_id=88b16b6440684467b882xxx154d8&enabled=false",
    "previous": null,
    "next": null
  }
}
```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.6.2 Querying User Details

Function

This API is used to query detailed information about a specified user.

URI

- URI format
GET /v3/users/{user_id}
- URI parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.

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 with Security Administrator permissions or a user token.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H 'X-Auth-Token:$token' -X GET https://sample.domain.com/v3/users/43cbe5e77aaf4665bbb962062dc1fxxx
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
user	Yes	JSON object	User details.

- Description for the user format

Parameter	Mandatory	Type	Description
description	Yes	String	Description for a user.
domain_id	Yes	String	ID of the tenant that the user belongs to.
enabled	Yes	Boolean	Indicates whether the user is enabled. The value can be true or false . The default value is true .
id	Yes	String	ID of a user.

Parameter	Mandatory	Type	Description
links	Yes	JSON object	Links of a user resource.
name	Yes	String	Username.
password_expires_at	Yes	String	UTC time when the password will expire. null indicates that the password has unlimited validity.
pwd_status	No	Boolean	Password status. true means that the password needs to be changed, and false means that the password is normal.
pwd_strength	No	String	Password strength. The value can be high , mid , or low .
default_project_id	No	String	ID of the project that is displayed by default when the user logs in to the console.
last_project_id	No	String	ID of the project that the user lastly accessed before exiting the system.

- Example response

```
{
  "users": [{
    "name": "username",
    "links": {
      "self": "https://sample.domain.com/v3/users/6d8b04e3bf99445b8f76300xxx"
    },
    "description": "1234",
    "domain_id": "88b16b6440684467b8825d7xxx",
    "enabled": false,
    "id": "6d8b04e3bf99445b8f763009xxx",
    "email": "",
    "default_project_id": "263fd9",
    "password_expires_at": "2016-12-07T00:00:00.000000Z",
    "pwd_status": true,
    "pwd_strength": "high",
    "last_project_id": ""
  }],
  "links": {
    "self": "https://sample.domain.com/v3/users?domain_id=88b16b6440684467b882xxx154d8&enabled=false",
    "previous": null,
    "next": null
  }
}
```

Status Codes

Status Code	Description
200	The request is successful.

Status Code	Description
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.6.3 Querying User Details (Recommended)

Function

This API can be used by the administrator to query the details about a specified user or used by a user to query their details.

URI

GET /v3.0/OS-USER/users/{user_id}

Table 4-29 URI parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.

Request Parameters

Table 4-30 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	A token with Security Administrator permissions is required if the administrator is requesting to query the details about a specified user. If an IAM user is requesting to query their details, the user token (no special permission requirements) of the user is required.

Response Parameters

Table 4-31 Parameters in the response body

Parameter	Type	Description
user	Object	User information.

Table 4-32 user

Parameter	Type	Description
enabled	Boolean	Enabling status of the user. true (default value) indicates that the user is enabled. false indicates that the user is disabled.
id	String	User ID.
domain_id	String	ID of the account to which the user belongs.
name	String	Username.
links	Object	User resource link information.
xuser_id	String	ID of the user in the external system.
xuser_type	String	Type of the user in the external system.
areacode	String	Country code.
email	String	Email address.
phone	String	Mobile number.

Parameter	Type	Description
pwd_status	Boolean	Password status. true means that the password needs to be changed, and false means that the password is normal.
update_time	String	Time when the user was last updated.
create_time	String	Time when the user was created.
last_login_time	String	Last login time of the user.
pwd_strength	String	Password strength. The value can be Low, Middle, High, or None .
is_domain_owner	Boolean	Indicates whether the user is the account administrator.
description	String	Description about the user.

Table 4-33 user.links

Parameter	Type	Description
self	String	Resource link.
previous	String	Previous resource link.
next	String	Next resource link.

Example Request

```
GET https://sample.domain.com/v3.0/OS-USER/users/{user_id}
```

Example Response

Status code: 200

The request is successful.

```
{
  "id": "",
  "xuser_type": "",
  "email": "",
  "user": {
    "pwd_strength": "Strong",
    "create_time": "2020-07-08 02:19:03.0",
    "last_login_time": null,
    "areacode": "",
    "enabled": true,
    "domain_id": "086ba757f90089cf0fe5c000dbe7f...",
    "xuser_id": "",
    "pwd_status": false,
    "update_time": null,
    "phone": "-",
    "name": "autotest1",
  }
}
```

```

"links" : {
  "next" : null,
  "previous" : null,
  "self" : "https://sample.domain.com/v3.0/OS-USER/users/093f75808b8089ba1f6dc000c7cac..."
},
"id" : "093f75808b8089ba1f6dc000c7cac...",
"xuser_type" : "",
"email" : "",
}
}

```

Status Codes

Status Code	Description
200	The request is successful.
403	Access denied.
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
500	Internal server error.

4.6.4 Querying the User Group to Which a User Belongs

Function

This API is used to query the information about the user group to which a specified user belongs.

URI

- URI format
GET /v3/users/{user_id}/groups
- URI parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.

Request Parameters

- Parameters in the request header

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

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission or authenticated token of the user.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token: $token" -X GET https://sample.domain.com/v3/users/43cbe5e77aaf4665bbb962062dc1fc9d/groups
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
groups	Yes	JSONArray	List of a user group.
links	Yes	JSON object	User group resource link.

- Description for the group format

Parameter	Mandatory	Type	Description
description	Yes	String	Description for a user group.
id	Yes	String	User group ID.
domain_id	Yes	String	ID of the domain where a user group is located.
name	Yes	String	User group name.
links	Yes	JSON object	User group resource link.
create_time	Yes	Long	Time when a user group is created.

- Example response

```
{
  "links": {
    "self": "https://sample.domain.com/v3/users/f7cb4876e5174c0885433e280e831c43/groups",
    "previous": null,
    "next": null
  },
  "groups": [{
    "description": "User group that has the permission for all system operations",
    "links": {
      "self": "https://sample.domain.com/v3/groups/e21c7a1e415c4604927948dc24750716"
    },
    "id": "e21c7a1e415c4604927948dc24750716",
    "create_time": 1472888495993,
    "domain_id": "88b16b6440684467b8825d7d96e154d8",
    "name": "admin"
  }
}
```



```
}  
}
```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.6.5 Querying Users in a User Group

Function

This API is used to query users in a user group.

URI

- URI format
GET /v3/groups/{group_id}/users
- URI parameters

Parameter	Mandatory	Type	Description
group_id	Yes	String	ID of a user group.

- Query parameters

Parameter	Mandatory	Type	Description
domain_id	No	String	ID of the domain to which a user group belongs.

Parameter	Mandatory	Type	Description
name	No	String	Name of a user. The maximum length is 64 characters.
enabled	No	String	Whether a user is enabled. The value can be true or false . true indicates the user is enabled and false indicates the user is not enabled.
password_expires_at	No	String	<p>Password expiration time. The format is password_expires_at=operator:timestamp.</p> <p>Example: password_expires_at=lt:2016-12-08T22:02:00Z</p> <ul style="list-style-type: none"> The value of operator can be lt, lte, gt, gte, eq, or neq. <ul style="list-style-type: none"> lt: The expiration time is earlier than <i>timestamp</i>. lte: The expiration time is earlier than or equal to <i>timestamp</i>. gt: The expiration time is later than <i>timestamp</i>. gte: The expiration time is equal to or later than <i>timestamp</i>. eq: The expiration time is equal to <i>timestamp</i>. neq: The expiration time is not equal to <i>timestamp</i>. The timestamp format is YYYY-MM-DDTHH:mm:ssZ.

Request Parameters

- Parameters in the request header

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

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/groups/00007111583e457389b0d4252643181b/users
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
links	Yes	JSON object	User resource link of a user group.
users	Yes	JSONArray	List of users in a user group.

- Description for the user format

Parameter	Mandatory	Type	Description
description	Yes	String	Description for a user.
domain_id	Yes	String	ID of the tenant that the user belongs to.
enabled	Yes	Boolean	Indicates whether the user is enabled. The value can be true or false . The default value is true .
id	Yes	String	User ID.
links	Yes	JSON object	User resource link.
name	Yes	String	Username.
password_expires_at	Yes	String	UTC time when the password will expire. null indicates that the password will not expire.
pwd_status	No	Boolean	Password status. true means that the password needs to be changed, and false means that the password is normal.
pwd_strength	No	String	Password strength. The value can be high , mid , or low .

Parameter	Mandatory	Type	Description
default_project_id	No	String	ID of the project that is displayed by default when the user logs in to the console.
last_project_id	No	String	ID of the project that the user lastly accessed before exiting the system.
email	No	String	User email address.

- Example response

```
{
  "users": [{
    "name": "username",
    "links": {
      "self": "https://sample.domain.com/v3/users/6d8b04e3bf99445b8f76300xxx"
    },
    "description": "1234",
    "domain_id": "88b16b6440684467b8825d7xxx",
    "enabled": false,
    "id": "6d8b04e3bf99445b8f763009xxx",
    "email": "",
    "default_project_id": "263fd9",
    "password_expires_at": "2016-12-07T00:00:00.000000Z",
    "pwd_status": true,
    "pwd_strength": "high",
    "last_project_id": ""
  }],
  "links": {
    "self": "https://sample.domain.com/v3/users?domain_id=88b16b6440684467b882xxx154d8&enabled=false",
    "previous": null,
    "next": null
  }
}
```

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.

4.6.6 Creating a User

Function

This API is used to create a user under a domain.

URI

POST /v3/users

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
name	Yes	String	A username with 5 to 32 characters. The username can contain special characters, but only hyphens (-), underscores (_), and spaces are allowed. It cannot start with a digit.
domain_id	No	String	ID of the domain where a user is located.
enabled	No	Boolean	Whether a user is enabled. The value can be true or false . true indicates the user is enabled and false indicates the user is not enabled. The default value is true .

Parameter	Mandatory	Type	Description
password	No	String	Password of the user. The password must meet the following requirements: <ul style="list-style-type: none"> • Must contain at least two of the following character types: uppercase letters, lowercase letters, digits, and special characters. • Cannot be the username or the username spelled backwards. • Cannot contain the user's mobile phone number or email address. • Must comply with the password policies under Account Settings.
default_project_id	No	String	Default project ID of a user.
description	No	String	Description of the user.

- **Example request**

1. Create the temporary file `${filename}.json` based on the following template. `${filename}` indicates the temporary file name, which is user-defined.

```
{
  "user": {
    "default_project_id": "acf2ffabba974fae8f30378ffde2cfa6",
    "domain_id": "88b16b6440684467b8825d7d96e154d8",
    "enabled": true,
    "name": "jamesdoe",
    "password": "*****"
  }
}
```

2. Run the following command under the directory storing the `${filename}.json` file.

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X POST -d @${filename}.json https://sample.domain.com/v3/users
```

3. Run the following command under the directory of the `${filename}.json` file to delete the `${filename}.json` file.

```
rm ${filename}.json
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
user	Yes	JSON object	User object.

- Description for the user format

Parameter	Mandatory	Type	Description
enabled	Yes	Boolean	Whether a user is enabled. The value can be true or false . true indicates the user is enabled and false indicates the user is not enabled. The default value is true .
id	Yes	String	User ID.
domain_id	Yes	String	ID of the domain where a user is located.
name	Yes	String	Username.
links	Yes	JSON object	User resource link.
default_project_id	No	String	Default project ID of a user.
password_expires_at	Yes	String	UTC when the password will expire. null indicates that the password will not expire.

- Example response

```
{
  "user": {
    "name": "jamesdoe",
    "links": {
      "self": "https://sample.domain.com/v3/users/614d1d2fb86940faab8f350bf1b9dbac"
    },
    "domain_id": "88b16b6440684467b8825d7d96e154d8",
    "enabled": true,
    "id": "614d1d2fb86940faab8f350bf1b9dbac",
    "default_project_id": "acf2ffabba974fae8f30378ffde2cfa6",
    "password_expires_at": null
  }
}
```

Status Codes

Status Code	Description
201	The user is successfully created.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.

Status Code	Description
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
409	A resource conflict occurs.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.6.7 Changing a Password

Function

This API is used to change the password for a user.

URI

- URI format
POST /v3/users/{user_id}/password
- URI parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.

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.

- Parameters in the request body

Parameter	Mandatory	Type	Description
original_password	Yes	String	Original password of a user.

Parameter	Mandatory	Type	Description
password	Yes	String	<p>User password after the change. The password must meet the following requirements:</p> <ul style="list-style-type: none"> • Must contain at least two of the following character types: uppercase letters, lowercase letters, digits, and special characters. • Cannot be the username or the username spelled backwards. • Cannot contain the user's mobile phone number or email address. • Must meet the requirements of the password policy configured on the account settings page.

- **Example request**

1. Create the temporary file ``${filename}.json`` based on the following template. ``${filename}`` indicates the temporary file name, which is user-defined.

```
{
  "user": {
    "password": "*****",
    "original_password": "*****"
  }
}
```

2. Run the following command under the directory storing the ``${filename}.json`` file.

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token: $token" -X POST -d @`${filename}.json` https://sample.domain.com/v3/users/2c1c6c54e59141b889c99e6fada5f19f/password
```

3. Run the following command under the directory of the ``${filename}.json`` file to delete the ``${filename}.json`` file.

```
rm `${filename}.json`
```

Response Parameters

None

Status Codes

Status Code	Description
204	The password is changed successfully.

Status Code	Description
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.6.8 Modifying User Information

Function

This API is used to modify user information under a domain.

URI

- URI format
PATCH /v3/users/{user_id}
- URI parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
name	No	String	A username with 5 to 32 characters. The username can contain special characters, but only hyphens (-), underscores (_), and spaces are allowed. It cannot start with a digit.
domain_id	No	String	ID of the domain where a user is located.
enabled	No	Boolean	Enabling status of the user. true indicates that the user is enabled. false indicates that the user is disabled. The default value is true .
password	No	String	User password after the change. The password must meet the following requirements: <ul style="list-style-type: none"> Must contain at least two of the following character types: uppercase letters, lowercase letters, digits, and special characters. Cannot be the username or the username spelled backwards. Cannot contain the user's mobile phone number or email address. Must comply with the password policies under Account Settings.
default_project_id	No	String	Default project ID of a user.
description	No	String	Description of the user.

- Example request

1. Create the temporary file `${filename}.json` based on the following template. `${filename}` indicates the temporary file name, which is user-defined.

```
{
  "user": {
    "name": "james1234",
    "default_project_id": "88b16b6440684467b8825d7d96e154d8",
    "enabled": false,
    "password": "*****"
  }
}
```

2. Run the following command under the directory storing the `${filename}.json` file.

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token: $token" -X POST -d @${filename}.json https://sample.domain.com/v3/users/2c1c6c54e59141b889c99e6fada5f19f
```

3. Run the following command under the directory of the `${filename}.json` file to delete the `${filename}.json` file.

```
rm ${filename}.json
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
user	Yes	JSON object	User object.

- Description for the user format

Parameter	Mandatory	Type	Description
enabled	Yes	Boolean	Whether a user is enabled. The value can be true or false . true indicates the user is enabled and false indicates the user is not enabled. The default value is true .
id	Yes	String	User ID.
domain_id	Yes	String	ID of the domain where a user is located.
name	Yes	String	Username.
links	Yes	JSON object	User resource link.
description	Yes	String	Description of the user.
default_project_id	No	String	Default project ID of a user.
password_expires_at	Yes	String	UTC when the password will expire. null indicates that the password will not expire.

- Example response

```
{
  "user": {
    "name": "james1234",
    "links": {
      "self": "https://sample.domain.com/v3/users/6d8b04e3bf99445b8f76300903e5bf32"
    },
    "description": {
    },
    "domain_id": "88b16b6440684467b8825d7d96e154d8",
    "enabled": false,
    "id": "6d8b04e3bf99445b8f76300903e5bf32",
    "default_project_id": "88b16b6440684467b8825d7d96e154d8",
    "password_expires_at": "2016-12-07T00:00:00.000000Z"
  }
}
```

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.
405	The method specified in the request is not allowed for the requested resource.
409	A resource conflict occurs.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.6.9 Modifying User Information (Including Email Address and Mobile Number)

Function

This API is provided for IAM users to modify their information.

URI

PUT /v3.0/OS-USER/users/{user_id}/info

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	User token.

- Parameters in the request body

Parameter	Mandatory	Type	Description
email	No	String	Email address with a maximum of 255 characters.
phone	No	String	Mobile number with a maximum of 32 digits. To change the mobile number, you must specify the country code.
areacode	No	String	Country code. This parameter is mandatory if the mobile number is changed.

- Example request

- Create a temporary file named `${filename}.json` using the following content format:

```
{
  "user": {
    "areacode": "0086",
    "phone": "18292010816",
    "email": "abcdefg@123.com"
  }
}
```

- Run the following command under the directory storing the `${filename}.json` file:
`curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H 'X-Auth-Token:$token' -X PUT -d @${filename}.json https://sample.domain.com/v3.0/OS-USER/users/0638848aa7801dbe1f01c01e92b95df7/info`
 - Run the following command to delete the `${filename}.json` file:
`rm ${filename}.json`

Response Parameters

None

Status Codes

Status Code	Description
204	The user information is modified successfully.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
409	A resource conflict occurs.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.6.10 Modifying User Information (Including Email Address and Mobile Number)

Function

This API is provided for the administrator to modify user information.

URI

PUT /v3.0/OS-USER/users/{user_id}

Table 4-34 URI parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.

Request Parameters

Table 4-35 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	Token with Security Administrator permissions.

Table 4-36 Parameters in the request body

Parameter	Mandatory	Type	Description
user	Yes	Object	IAM user information.

Table 4-37 user

Parameter	Mandatory	Type	Description
name	No	String	New username with 5–32 characters. The username can contain special characters, but only hyphens (-), underscores (_), and spaces are allowed. It cannot start with a digit.
password	No	String	Password of the user. The password must meet the following requirements: <ul style="list-style-type: none"> • Can contain 6 to 32 characters. The default minimum password length is 6 characters. • Must contain at least two of the following character types: uppercase letters, lowercase letters, digits, and special characters. • Must meet the requirements of the password policy configured on the account settings page. • Must be different from the old password.
email	No	String	Email address, which can contain not more than 255 characters.

Parameter	Mandatory	Type	Description
areacode	No	String	Country code. The country code must be used together with a mobile number.
phone	No	String	New mobile number, which can contain a maximum of 32 digits. The mobile number must be used together with a country code.
enabled	No	Boolean	Enabling status of the IAM user. true (default value) indicates that the user is enabled. false indicates that the user is disabled.
pwd_status	No	Boolean	Indicates whether the user must change their password at the first login. true (default value) indicates that the user must change their password at the first login. false indicates that the user does not need to change their password at the first login.
xuser_type	No	String	Type of the user in the external system. The user type can contain a maximum of 64 characters. xuser_type must be used together with xuser_id and will be verified based on xaccount_type and xdomain_type of the same account. NOTE An external system refers to an enterprise management system connected to cloud system. Parameters xaccount_type , xaccount_id , xdomain_type , xdomain_id , xuser_type , and xuser_id cannot be obtained from the cloud system. Please contact the enterprise administrator.
xuser_id	No	String	ID of the user in the external system. The user ID can contain a maximum of 128 characters, and must be used together with xuser_type . NOTE An external system refers to an enterprise management system connected to cloud system. Parameters xaccount_type , xaccount_id , xdomain_type , xdomain_id , xuser_type , and xuser_id cannot be obtained from the cloud system. Please contact the enterprise administrator.

Parameter	Mandatory	Type	Description
description	No	String	Description of the IAM user.

Response Parameters

Table 4-38 Parameters in the response body

Parameter	Type	Description
user	Object	IAM user information.

Table 4-39 user

Parameter	Type	Description
pwd_status	Boolean	Whether password reset is required at first login.
xuser_id	String	ID of the user in the external system. NOTE An external system refers to an enterprise management system connected to cloud system. Parameters xaccount_type , xaccount_id , xdomain_type , xdomain_id , xuser_type , and xuser_id cannot be obtained from the cloud system. Please contact the enterprise administrator.
xuser_type	String	Type of the user in the external system. NOTE An external system refers to an enterprise management system connected to cloud system. Parameters xaccount_type , xaccount_id , xdomain_type , xdomain_id , xuser_type , and xuser_id cannot be obtained from the cloud system. Please contact the enterprise administrator.
description	String	Description of the IAM user.
name	String	New IAM user name with 5 to 32 characters. The username can contain special characters, but only hyphens (-), underscores (_), and spaces are allowed. It cannot start with a digit.
phone	String	New mobile number, which can contain a maximum of 32 digits. The mobile number must be used together with a country code.
domain_id	String	ID of the account to which the user belongs.

Parameter	Type	Description
enabled	Boolean	Enabling status of the IAM user. true (default value) indicates that the user is enabled. false indicates that the user is disabled.
pwd_status	Boolean	Indicates whether the user must change their password at the first login. true (default value) indicates that the user must change their password at the first login. false indicates that the user does not need to change their password at the first login.
areacode	String	Country code.
email	String	New email address.
id	String	IAM user ID.
links	Object	User resource link information.
password_expires_at	String	UTC time when the password will expire. null indicates that the password has unlimited validity.

Table 4-40 user.links

Parameter	Type	Description
self	String	Resource link.

Example Request

```
PUT https://sample.domain.com/v3.0/OS-USER/users/{user_id}
{
  "user": {
    "email": "IAMEmail@123.com",
    "areacode": "0086",
    "phone": "12345678910",
    "enabled": true,
    "name": "IAMUser",
    "password": "IAMPassword@",
    "pwd_status": false,
    "xuser_type": "",
    "xuser_id": "",
    "description": "IAMDescription"
  }
}
```

Example Response

Status code: 200

The request is successful.

```
{
  "user": {
```

```

    "description": "IAMDescription",
    "areacode": "0086",
    "enabled": true,
    "pwd_status": false,
    "xuser_id": "",
    "domain_id": "d78cbac186b744899480f25bd0...",
    "phone": "12345678910",
    "name": "IAMUser",
    "links": {
      "self": "https://sample.domain.com/3.0/OS-USER/users/076934ff9f0010cd1f0bc003..."
    },
    "id": "076934ff9f0010cd1f0bc0031019...",
    "xuser_type": "",
    "email": "IAMEmail@123.com"
  }
}

```

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.
405	The method specified in the request is not allowed for the requested resource.
409	A resource conflict occurs.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

Error Codes

Status Code	Error Code	Error Message
400	1100	Mandatory parameters are missing.
400	1101	Invalid username.
400	1102	Invalid email address.
400	1103	Incorrect password.
400	1104	Invalid mobile number.

Status Code	Error Code	Error Message
400	1105	The value of xuser_type must be the same as that of xdomain_type .
400	1106	The country code and mobile number must be set at the same time.
400	1107	The account administrator cannot be deleted.
400	1108	The new password must be different from the old password.
400	1109	The username already exists.
400	1110	The email address has already been used.
400	1111	The mobile number has already been used.
400	1113	The user ID or user type already exists.
400	1115	The number of IAM users has reached the maximum allowed limit.
400	1117	Invalid user description.

4.6.11 Deleting a User

Function

This API is used to delete a user.

URI

- URI format
DELETE /v3/users/{user_id}
- URI parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H "X-Auth-Token:$token" -X DELETE https://sample.domain.com/v3/users/2c1c6c54e59141b889c99e6fada5f19f
```

Response Parameters

None

Status Codes

Status Code	Description
204	The user is deleted successfully.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.6.12 Deleting a User from a User Group

Function

This API is used to delete a user from a user group.

URI

- URI format
DELETE /v3/groups/{group_id}/users/{user_id}
- URI parameters

Parameter	Mandatory	Type	Description
group_id	Yes	String	ID of a user group.
user_id	Yes	String	ID of a user.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H "X-Auth-Token: $token" -X DELETE https://sample.domain.com/v3/groups/00007111583e457389b0d4252643181b/users/edb66d2b656c43d0b67fb143d670bb3a
```

Response Parameters

None

Status Codes

Status Code	Description
204	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.

4.6.13 Querying MFA Device Information of Users

Function

This API is provided for the administrator to query the MFA device information of users.

URI

GET /v3.0/OS-MFA/virtual-mfa-devices

Request Parameters

Table 4-41 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

Table 4-42 Parameters in the response body

Parameter	Type	Description
virtual_mfa_devices	Array of objects	Virtual MFA device information.

Table 4-43 virtual_mfa_devices

Parameter	Type	Description
serial_number	String	Virtual MFA device serial number.
user_id	String	User ID.

Example Request

```
GET https://sample.domain.com/v3.0/OS-MFA/virtual-mfa-devices
```

Example Response

Status code: 200

The request is successful.


```
{
  "virtual_mfa_devices" : [
    {
      "user_id" : "16b26081f43d4c628c4bb88cf32e9...",
      "serial_number" : "iam/mfa/16b26081f43d4c628c4bb88cf32e9..."
    },
    {
      "user_id" : "47026081f43d4c628c4bb88cf32e9...",
      "serial_number" : "iam/mfa/75226081f43d4c628c4bb88cf32e9..."
    }
  ]
}
```

Status code: 403

Access denied.

- Example 1

```
{
  "error_msg" : "You are not authorized to perform the requested action.",
  "error_code" : "IAM.0002"
}
```

- Example 2

```
{
  "error_msg" : "Policy doesn't allow %(actions)s to be performed.",
  "error_code" : "IAM.0003"
}
```

Status code: 404

The requested resource cannot be found.

```
{
  "error_msg" : "Could not find %(target)s: %(target_id)s.",
  "error_code" : "IAM.0004"
}
```

Status code: 500

Internal server error.

```
{
  "error_msg" : "An unexpected error prevented the server from fulfilling your request.",
  "error_code" : "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.6.14 Querying the MFA Device Information of a User

Function

This API can be used by the administrator to query the MFA device information of a specified user or used by a user to query their MFA device information.

URI

GET /v3.0/OS-MFA/users/{user_id}/virtual-mfa-device

Table 4-44 URI parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.

Request Parameters

Table 4-45 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	A token with Security Administrator permissions is required if the administrator is requesting to query the MFA device information of a specified user. If a user is requesting to query their MFA device information, the user token (no special permission requirements) of the user is required.

Response Parameters

Table 4-46 Parameters in the response body

Parameter	Type	Description
virtual_mfa_device	object	Virtual MFA device information.

Table 4-47 virtual_mfa_device

Parameter	Type	Description
serial_number	String	Virtual MFA device serial number.
user_id	String	User ID.

Example Request

```
GET https://sample.domain.com/v3.0/OS-MFA/users/{user_id}/virtual-mfa-device
```

Example Response

Status code: 200

The request is successful.

```
{
  "virtual_mfa_device" :
  {
    "user_id" : "16b26081f43d4c628c4bb88cf32e9...",
    "serial_number" : "iam/mfa/16b26081f43d4c628c4bb88cf32e9..."
  }
}
```

Status code: 403

Access denied.

- Example 1

```
{
  "error_msg" : "You are not authorized to perform the requested action.",
  "error_code" : "IAM.0002"
}
```

- Example 2

```
{
  "error_msg" : "Policy doesn't allow %(actions)s to be performed.",
  "error_code" : "IAM.0003"
}
```

Status code: 404

The requested resource cannot be found.

```
{
  "error_msg" : "Could not find %(target)s: %(target_id)s.",
  "error_code" : "IAM.0004"
}
```

Status code: 500

Internal server error.

```
{
  "error_msg" : "An unexpected error prevented the server from fulfilling your request.",
  "error_code" : "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.6.15 Querying Login Protection Configurations of Users

Function

This API is provided for the administrator to query the login protection configurations of users.

URI

GET /v3.0/OS-USER/login-protects

Request Parameters

Table 4-48 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

Table 4-49 Parameters in the response body

Parameter	Type	Description
login_protects	Array of objects	Login protection configurations. NOTE The response only includes the login protection configurations of users for whom login protection has been configured.

Table 4-50 login_protects

Parameter	Type	Description
enabled	Boolean	Indicates whether login protection has been enabled for a user. The value can be true or false .
user_id	String	User ID.
verification_method	String	Login authentication method of the user. <ul style="list-style-type: none"> ● email: email verification code ● vmfa: virtual MFA verification code ● SMS: SMS verification code

Example Request

GET https://sample.domain.com/v3.0/OS-USER/login-protects

Example Response

Status code: 200

The request is successful.

```
{
  "login_protects" : [
    {
      "user_id" : "75226081f43d4c628c4bb88cf32e9...",
      "enabled" : true,
      "verification_method" : "email"
    },
    {
      "user_id" : "16b26081f43d4c628c4bb88cf32e9...",
      "enabled" : true,
      "verification_method" : "vmfa"
    },
    {
      "user_id" : "56b26081f43d4c628c4bb88cf32e9...",
      "enabled" : true,
      "verification_method" : "sms"
    },
    {
      "user_id" : "08c16cb6c58010691f81c0028dd94...",
      "enabled" : false,
      "verification_method" : "none"
    }
  ]
}
```

NOTE

If login protection has never been configured for a user, you cannot use this API to obtain the login protection configuration of the user.

Status code: 403

Access denied.

- Example 1

```
{
  "error_msg": "You are not authorized to perform the requested action.",
  "error_code": "IAM.0002"
}
```

- Example 2

```
{
  "error_msg": "Policy doesn't allow %(actions)s to be performed.",
  "error_code": "IAM.0003"
}
```

Status code: 404

The requested resource cannot be found.

```
{
  "error_msg": "Could not find %(target)s: %(target_id)s.",
  "error_code": "IAM.0004"
}
```

Status code: 500

Internal server error.

```
{
  "error_msg": "An unexpected error prevented the server from fulfilling your request.",
  "error_code": "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.6.16 Querying the Login Protection Configuration of a User

Function

This API can be used by the administrator to query the login protection configuration of a specified user or used by a user to query their login protection configuration.

URI

GET /v3.0/OS-USER/users/{user_id}/login-protect

Table 4-51 URI parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.

Request Parameters

Table 4-52 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	A token with Security Administrator permissions is required if the administrator is requesting to query the login protection configuration of a specified user. If a user is requesting to query their login protection configuration, the user token (no special permission requirements) of the user is required.

Response Parameters

Status code: 200

Table 4-53 Parameters in the response body

Parameter	Type	Description
login_protect	object	Login protection configuration.

Table 4-54 login_protect

Parameter	Type	Description
enabled	Boolean	Indicates whether login protection has been enabled for a user. The value can be true or false .
user_id	String	User ID.
verification_method	String	Login authentication method of the user.

Example Request

```
GET https://sample.domain.com/v3.0/OS-USER/users/{user_id}/login-protect
```

Example Response

Status code: 200

The request is successful.

```
{
  "login_protect" : {
    "user_id" : "16b26081f43d4c628c4bb88cf32e9...",
    "enabled" : true,
    "verification_method" : "vmfa"
  }
}
```

Status code: 403

Access denied.

- Example 1

```
{
  "error_msg" : "You are not authorized to perform the requested action.",
  "error_code" : "IAM.0002"
}
```

- Example 2

```
{
  "error_msg" : "Policy doesn't allow %(actions)s to be performed.",
  "error_code" : "IAM.0003"
}
```

Status code: 404

The requested resource cannot be found.

```
{
  "error_msg" : "Could not find %(target)s: %(target_id)s.",
  "error_code" : "IAM.0004"
}
```

NOTE

If login protection has never been configured for a user, you cannot use this API to obtain the login protection configuration of the user. Otherwise, the error code IAM.0004 will be returned.

Status code: 500

Internal server error.

```
{
  "error_msg" : "An unexpected error prevented the server from fulfilling your request.",
  "error_code" : "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.

Status Code	Description
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.6.17 Creating a Virtual MFA Device

Function

This API is provided for IAM users to create a virtual MFA device.

URI

POST /v3.0/OS-MFA/virtual-mfa-devices

Request Parameters

Table 4-55 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token (no special permission requirements) of the IAM user corresponding to the user_id specified in the request body.

Table 4-56 Parameters in the request body

Parameter	Mandatory	Type	Description
virtual_mfa_device	Yes	object	MFA device information.

Table 4-57 virtual_mfa_device

Parameter	Mandatory	Type	Description
name	Yes	String	Device name. Minimum length: 1 character Maximum length: 64 characters

Parameter	Mandatory	Type	Description
user_id	Yes	String	ID of the user for whom you will create the MFA device.

Response Parameters

Status code: 201

Table 4-58 Parameters in the response body

Parameter	Type	Description
virtual_mfa_device	object	MFA device information.

Table 4-59 virtual_mfa_device

Parameter	Type	Description
serial_number	String	Serial number of the MFA device.
base32_string_seed	String	Base32 seed, which a third-party system can use to generate a CAPTCHA code.

Example Request

POST https://sample.domain.com/v3.0/OS-MFA/virtual-mfa-devices

```
{
  "virtual_mfa_device": {
    "name": "{device_name}",
    "user_id": "09f99d8f6a001d4f1f01c00c31968..."
  }
}
```

Example Response

Status code: 201

The request is successful.

```
{
  "virtual_mfa_device": {
    "serial_number": "iam:09f6bd6a96801de40f01c00c85691...:mfa/{device_name}",
    "base32_string_seed": "{string}"
  }
}
```

Status Codes

Status Code	Description
201	The request is successful.
400	The request is invalid.
401	Authentication failed.
403	You do not have permission to perform this action.
409	A conflict occurs when the requested resource is saved.
500	A system error occurred.

4.6.18 Deleting a Virtual MFA Device

Function

This API is provided for the administrator to delete their own virtual MFA device.

URI

DELETE /v3.0/OS-MFA/virtual-mfa-devices

Table 4-60 Query parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	ID of the user whose virtual MFA device is to be deleted, that is, the administrator's user ID.
serial_number	Yes	String	Serial number of the virtual MFA device.

Request Parameters

Table 4-61 Parameters in the request header

Parameter	Mandatory	Type	Description
X-auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

None

Example Request

```
DELETE https://sample.domain.com/v3.0/OS-MFA/virtual-mfa-devices?
user_id=09f6bd85fc801de41f0cc00ce9172...&serial_number=iam:09f6bd6a96801de40f01c00c85691...:mfa/
{device_name}
```

Example Response

None

Status Codes

Status Code	Description
204	The request is successful.
401	Authentication failed.
403	You do not have permission to perform this action.
500	A system error occurred.

4.6.19 Binding a Virtual MFA Device

Function

This API is provided for IAM users to bind a virtual MFA device.

URI

PUT /v3.0/OS-MFA/mfa-devices/bind

Request Parameters

Table 4-62 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-token	Yes	String	Token (no special permission requirements) of the IAM user corresponding to the user_id specified in the request body.

Table 4-63 Parameters in the request body

Parameter	Mandatory	Type	Description
user_id	Yes	String	ID of the user to whom you will bind the virtual MFA device.
serial_number	Yes	String	Serial number of the virtual MFA device.
authentication_code_first	Yes	String	Verification code 1.
authentication_code_second	Yes	String	Verification code 2.

Response Parameters

None

Example Request

```
PUT https://sample.domain.com/v3.0/OS-MFA/mfa-devices/bind
{
  "user_id" : "09f99d8f6a001d4f1f01c00c31968...",
  "authentication_code_first" : "977931",
  "authentication_code_second" : "527347",
  "serial_number" : "iam:09f6bd6a96801de40f01c00c85691...:mfa/{device_name}"
}
```

Example Response

None

Status Codes

Status Code	Description
204	The request is successful.
400	The request is invalid.
401	Authentication failed.
403	You do not have permission to perform this action.
404	The requested resource cannot be found.
409	A conflict occurs when the requested resource is saved.
500	A system error occurred.

4.6.20 Unbinding a Virtual MFA Device

Function

This API is used by the administrator to unbind a virtual MFA device from an IAM user, or used by an IAM user to unbind their own virtual MFA device.

URI

PUT /v3.0/OS-MFA/mfa-devices/unbind

Request Parameters

Table 4-64 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	<ul style="list-style-type: none"> Administrator: Provide a token with Security Administrator permissions. User: Provide the token (no special permission requirements) of the user specified in user_id of the request body.

Table 4-65 Parameters in the request body

Parameter	Mandatory	Type	Description
user_id	Yes	String	ID of the user from whom you will unbind the MFA device.
authentication_code	Yes	String	<ul style="list-style-type: none"> Administrator: Set this parameter to any value, because verification is not required. IAM user: Enter the MFA verification code.
serial_number	Yes	String	Serial number of the MFA device.

Response Parameters

None

Example Request

```
PUT https://sample.domain.com/v3.0/OS-MFA/mfa-devices/unbind
```

```
{
  "user_id" : "09f99d8f6a001d4f1f01c00c31968...",
  "authentication_code" : "373658",
  "serial_number" : "iam:09f6bd6a96801de40f01c00c85691...mfa/{device_name}"
}
```

Example Response

None

Status Codes

Status Code	Description
204	The request is successful.
400	The request is invalid.
401	Authentication failed.
403	You do not have permission to perform this action.
404	The requested resource cannot be found.
409	A conflict occurs when the requested resource is saved.
500	A system error occurred.

4.6.21 Modifying the Login Protection Configuration of a User

Function

This API is provided for the administrator to modify the login protection configuration of a user.

URI

PUT /v3.0/OS-USER/users/{user_id}/login-protect

Table 4-66 URI parameters

Parameter	Mandatory	Type	Description
user_id	Yes	String	ID of the user whose login protection configuration is to be modified.

Request Parameters

Table 4-67 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-token	Yes	String	Token with Security Administrator permissions.

Table 4-68 Parameters in the request body

Parameter	Mandatory	Type	Description
login_protect	Yes	object	Login protection configuration.

Table 4-69 Login_project

Parameter	Mandatory	Type	Description
enabled	Yes	Boolean	Indicates whether login protection has been enabled for the user. The value can be true or false .
verification_method	Yes	String	Login authentication method of the user. Options: sms , email , and vmfa .

Response Parameters

Status code: 200

Table 4-70 Parameters in the response body

Parameter	Type	Description
login_protect	object	Login protection configuration.

Table 4-71 login_protect

Parameter	Type	Description
user_id	String	User ID.

Parameter	Type	Description
enabled	Boolean	Indicates whether login protection has been enabled for the user. The value can be true or false .
verification_method	String	Login authentication method of the user. Options: sms , email , and vmfa .

Example Request

```
PUT https://sample.domain.com/v3.0/OS-USER/users/{user_id}/login-protect
{
  "login_protect": {
    "enabled": true,
    "verification_method": "vmfa"
  }
}
```

Example Response

Status code: 200

The request is successful.

```
{
  "login_protect": {
    "user_id": "16b26081f43d4c628c4bb88cf32e9...",
    "enabled": true,
    "verification_method": "vmfa"
  }
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The request is invalid.
401	Authentication failed.
403	You do not have permission to perform this action.
404	The requested resource cannot be found.
500	A system error occurred.

4.7 User Group Management

4.7.1 Listing User Groups

Function

This API is used to query user group information.

URI

- URI format
GET /v3/groups{?domain_id,name}
- Query parameters

Parameter	Mandatory	Type	Description
domain_id	No	String	ID of the domain where a user group is located.
name	No	String	Name of a user group. The length is less than or equal to 64 characters.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://sample.domain.com/v3/groups?domain_id=ac7197fd67a24dc5850972854729a762&name=group123
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
links	Yes	JSON object	User group resource link.
groups	Yes	JSONArray	List of a user group.

- Group parameter description

Parameter	Mandatory	Type	Description
description	Yes	String	Description for a user group.
domain_id	Yes	String	ID of the domain to which a user group belongs.
id	Yes	String	ID of a user group.
links	Yes	JSON object	User group resource link.
name	Yes	String	Name of a user group.
create_time	Yes	Long	Time when a user group is created.

- Example response

```
{
  "links": {
    "self": "https://sample.domain.com/v3/groups?
domain_id=ac7197fd67a24dc5850972854729a762&name=group123",
    "previous": null,
    "next": null
  },
  "groups": [{
    "description": "",
    "links": {
      "self": "https://sample.domain.com/v3/groups/ff74abaeabe34c278a4b7693c7f0dff7"
    },
    "id": "ff74abaeabe34c278a4b7693c7f0dff7",
    "create_time": 1482566254983,
    "domain_id": "ac7197fd67a24dc5850972854729a762",
    "name": "group123"
  }]
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.

4.7.2 Querying User Group Details

Function

This API is used to query detailed information about a user group.

URI

- URI format
GET /v3/groups/{group_id}
- Query parameters

Parameter	Mandatory	Type	Description
group_id	Yes	String	ID of a user group.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://sample.domain.com/v3/groups/ab9f261180d746ef8624beb5ae39b5aa
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
group	Yes	JSON object	Response body of a user group.
description	Yes	String	Description for a user group.
domain_id	Yes	String	ID of the domain to which a user group belongs.
id	Yes	String	ID of a user group.
links	Yes	JSON object	Links to a user group.
name	Yes	String	Name of a user group.
create_time	Yes	Long	Time when a user group is created.

- Example response

```
{
  "group":{
    "domain_id":"d54061ebcb5145dd814f8eb3fe9b7ac0",
    "description":"Contract developers",
    "id":"ab9f261180d746ef8624beb5ae39b5aa",
```

```

"links":{
  "self":"https://sample.domain.com/v3/groups/ab9f261180d746ef8624beb5ae39b5aa"
},
"name":"abcdef",
"create_time": 1494943784468
}

```

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.

4.7.3 Creating a User Group

Function

This API is used to create a user group.

URI

POST /v3/groups

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
description	No	String	Description for a user group. The length is less than or equal to 255 characters.

Parameter	Mandatory	Type	Description
domain_id	No	String	ID of the domain to which a user group belongs.
name	Yes	String	Name of a user group. The length is less than or equal to 64 characters.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H 'X-Auth-Token:$token' -X POST -d '{"group": {"description": "Contract developers", "domain_id": "d54061ebcb5145dd814f8eb3fe9b7ac0", "name": "jixiang2"}}' https://sample.domain.com/v3/groups
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
description	Yes	String	Description for a user group.
domain_id	Yes	String	ID of the domain to which a user group belongs.
id	Yes	String	ID of a user group.
links	Yes	JSON object	Links to a user group.
name	Yes	String	Name of a user group.

- Example response

```
{
  "group":{
    "domain_id":"d54061ebcb5145dd814f8eb3fe9b7ac0",
    "description":"Contract developers",
    "id":"ab9f261180d746ef8624beb5ae39b5aa",
    "links":{
      "self":"https://sample.domain.com/v3/groups/ab9f261180d746ef8624beb5ae39b5aa"
    },
    "name":"abcdef"
  }
}
```

Status Codes

Status Code	Description
201	The user group is successfully created.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.

Status Code	Description
409	A resource conflict occurs.

4.7.4 Adding a User to a User Group

Function

This API is used to add a user to a user group.

URI

- URI format
PUT /v3/groups/{group_id}/users/{user_id}
- URI parameters

Parameter	Mandatory	Type	Description
group_id	Yes	String	ID of a user group.
user_id	Yes	String	ID of a user.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H "X-Auth-Token:$token" -X PUT https://sample.domain.com/v3/groups/00007111583e457389b0d4252643181b/users/edb66d2b656c43d0b67fb143d670bb3a
```

Response Parameters

None

Status Codes

Status Code	Description
204	The request is successful.

Status Code	Description
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.

4.7.5 Updating a User Group

Function

This API is used to update user group information.

URI

- URI format
PATCH /v3/groups/{group_id}
- URI parameters

Parameter	Mandatory	Type	Description
group_id	Yes	String	ID of a user group.

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
group	Yes	Object	Request body of a group.
description	No	String	Description for a user group. The length is less than or equal to 255 characters.

Parameter	Mandatory	Type	Description
domain_id	No	String	ID of the domain to which a user group belongs.
name	No	String	Name of a user group. The length is less than or equal to 64 characters.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H 'X-Auth-Token:$token' -X PATCH -d '{"group": {"description": "Contract developers 2016"}}' https://sample.domain.com/v3/groups/aaec2abd4eba430fbf61541ffde76650
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
group	Yes	Dict	Response body of a user group.
description	Yes	String	Description for a user group.
domain_id	Yes	String	ID of the domain to which a user group belongs.
id	Yes	String	ID of a user group.
links	Yes	Dict	User group resource link.
name	Yes	String	Name of a user group.

- Example response

```
{
  "group": {
    "domain_id": "d54061ebcb5145dd814f8eb3fe9b7ac0",
    "description": "Contract developers 2016",
    "id": "aaec2abd4eba430fbf61541ffde76650",
    "links": {
      "self": "https://sample.domain.com/v3/groups/aaec2abd4eba430fbf61541ffde76650"
    },
    "name": "jixiang1"
  }
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.

Status Code	Description
404	The requested resource cannot be found.
409	A resource conflict occurs.
501	The API is not implemented.

4.7.6 Deleting a User Group

Function

This API is used to delete a user group.

URI

- URI format
DELETE /v3/groups/{group_id}
- URI parameters

Parameter	Mandatory	Type	Description
group_id	Yes	String	ID of a user group.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H "X-Auth-Token: $token" -X DELETE https://sample.domain.com/v3/groups/aaec2abd4eba430bf61541ffde76650
```

Response Parameters

None

Status Codes

Status Code	Description
204	The user group is deleted successfully.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.

4.7.7 Querying Whether a User Belongs to a User Group

Function

This API is used to query whether a user belongs to a user group.

URI

- URI format
HEAD /v3/groups/{group_id}/users/{user_id}
- URI parameters

Parameter	Mandatory	Type	Description
group_id	Yes	String	ID of a user group.
user_id	Yes	String	ID of a user.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H "X-Auth-Token: $token" -X HEAD https://sample.domain.com/v3/groups/00007111583e457389b0d4252643181b/users/edb66d2b656c43d0b67fb143d670bb3a
```

Response Parameters

None

Status Codes

Status Code	Description
204	The user belongs to this user group.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
404	The server could not find the requested page, or the user does not belong to this user group.

4.8 Permission Management

4.8.1 Querying a Role List

Function

This API is used to query a role list, including the permissions policies of a role. A role is a set of permissions and represents a group of actions.

URI

GET /v3/roles

Request Parameters

Table 4-72 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	Token with Security Administrator permissions.

Response Parameters

Table 4-73 Parameters in the response body

Parameter	Type	Description
links	Object	Resource link information.
roles	Array of objects	Permission information.
total_number	Integer	Total number of permissions.

Table 4-74 links

Parameter	Type	Description
self	String	Resource link.
previous	String	Previous resource link.
next	String	Next resource link.

Table 4-75 roles

Parameter	Type	Description
domain_id	String	ID of the domain to which the permission belongs.
flag	String	If this parameter is set to fine_grained , the permission is a system-defined policy.
description_cn	String	Description of the permission in Chinese.
catalog	String	Service catalog of the permission.
name	String	Permission name. This parameter is carried in the token of a user. The cloud service determines whether the user has the access permission based on the role name.
description	String	Description of the permission.
links	Object	Permission resource link.
id	String	Permission ID.

Parameter	Type	Description
display_name	String	Display name of the permission.
type	String	Display mode of the permission. NOTE <ul style="list-style-type: none"> • AX: Account level. • XA: Project level. • AA: Both the account level and project level. • XX: Neither the account level nor project level. • The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.
policy	Object	Content of the permission.
updated_time	String	Time when the permission was last updated.
created_time	String	Time when the permission was created.

Table 4-76 roles.links

Parameter	Type	Description
self	String	Resource link.
previous	String	Previous resource link.
next	String	Next resource link.

Table 4-77 roles.policy

Parameter	Type	Description
Depends	Array of objects	Dependence permissions.
Statement	Array of objects	Statement of the permission.

Parameter	Type	Description
Version	String	Permission version. NOTE <ul style="list-style-type: none"> • 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. • 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.

Table 4-78 roles.policy.Depends

Parameter	Type	Description
catalog	String	Service catalog of the permission.
display_name	String	Display name of the permission.

Table 4-79 roles.policy.Statement

Parameter	Type	Description
Action	Array of strings	Specific operation permission on a resource. A maximum of 100 actions are allowed. NOTE <ul style="list-style-type: none"> • The value format is <i>Service name.Resource type.Operation</i>, for example, vpc:ports:create. • <i>Service name</i>: indicates the product name, such as ecs, evs, or vpc. Only lowercase letters are allowed. Resource types and operations are not case-sensitive. You can use an asterisk (*) to represent all operations. • For a custom agency policy, this parameter should be set to "Action": ["iam:agencies:assume"].
Effect	String	Effect of the permission. The value can be Allow or Deny . If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements. Options: <ul style="list-style-type: none"> • Allow • Deny
Condition	Object	Conditions for the permission to take effect. A maximum of 10 conditions are allowed.

Parameter	Type	Description
Resource	Array of strings	<p>Cloud resource. The array can contain a maximum of 10 resource strings, and each string cannot exceed 128 characters.</p> <p>NOTE</p> <ul style="list-style-type: none"> • Format: <code>region::bucket*</code>. Asterisks are allowed. • The region segment can be <code>*</code> or a region accessible to the user. The specified resource must belong to the corresponding service that actually exists. • In the case of a custom policy for agencies, the type of this parameter is object, and the value should be set to <code>"Resource": {"uri": ["/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"]}</code>.

Table 4-80 roles.policy.Statement.Condition.operator

Parameter	Type	Description
attribute	Array of strings	<p>Condition key. The condition key must correspond to the specified operator. A maximum of 10 condition keys are allowed.</p> <p>The parameter type is custom character string array.</p>

Example Request

```
GET https://sample.domain.com/v3/roles
```

Example Response

Status code: 200

The request is successful.

```
{
  "roles": [ {
    "domain_id": null,
    "description_cn": "Description of the permission in Chinese",
    "catalog": "VulnScan",
    "name": "wscn_admin",
    "description": "Vulnerability Scan Service administrator of tasks and reports.",
    "links": {
      "next": null,
      "previous": null,
      "self": "https://sample.domain.com/v3/roles/0af84c1502f447fa9c2fa18083fbb..."
    },
    "id": "0af84c1502f447fa9c2fa18083fbb...",
    "display_name": "VSS Administrator",
    "type": "XA",
    "policy": {
      "Version": "1.0",
      "Statement": [ {
```



```

    "Action" : [ "WebScan:*:*" ],
    "Effect" : "Allow"
  }],
  "Depends" : [ {
    "catalog" : "BASE",
    "display_name" : "Server Administrator"
  }, {
    "catalog" : "BASE",
    "display_name" : "Tenant Guest"
  } ]
}
}, {
  "domain_id" : null,
  "flag" : "fine_grained",
  "description_cn" : "Description of the permission in Chinese",
  "catalog" : "CSE",
  "name" : "system_all_34",
  "description" : "All permissions of CSE service.",
  "links" : {
    "next" : null,
    "previous" : null,
    "self" : "https://sample.domain.com/v3/roles/0b5ea44ebdc64a24a9c372b2317f7..."
  },
  "id" : "0b5ea44ebdc64a24a9c372b2317f7...",
  "display_name" : "CSE Admin",
  "type" : "XA",
  "policy" : {
    "Version" : "1.1",
    "Statement" : [ {
      "Action" : [ "cse:*:*", "ecs:*:*", "evs:*:*", "vpc:*:*" ],
      "Effect" : "Allow"
    } ]
  }
}],
"links" : {
  "next" : null,
  "previous" : null,
  "self" : "https://sample.domain.com/v3/roles"
},
"total_number" : 300
}

```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.

4.8.2 Querying Role Details

Function

This API is used to query role details, including the permissions policies of a role. A role is a set of permissions and represents a group of actions.

URI

- URI format
GET /v3/roles/{role_id}
- URI parameters

Parameter	Mandatory	Type	Description
role_id	Yes	String	ID of a role.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://sample.domain.com/v3/roles/19bb93eec4ca4f08aefdc02da76d8f3c
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
role	Yes	Dict	Details of the role.

- Description for the role format

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the domain to which a role belongs.
id	Yes	String	ID of a role.
links	Yes	Dict	Role resource link.
name	Yes	String	Name of a role.

Parameter	Mandatory	Type	Description
type	Yes	String	Display mode of a role. <ul style="list-style-type: none"> ● AX: A role is displayed at the domain layer. ● XA: A role is displayed at the project layer. ● AA: A role is displayed at both the domain and project layers. ● XX: A role is not displayed at the domain or project layer.
display_name	No	String	Displayed name of a role.
catalog	No	String	Directory where a role locates.
policy	No	Dict	Policy of a role.
description	No	String	Description of a role.

- Example response

```
{
  "role": {
    "display_name": "Tanent Guest",
    "description": "Tanent Guest",
    "links": {
      "self": "https://sample.domain.com/v3/roles/19bb93eec4ca4f08aefdc02da76d8f3c"
    },
    "domain_id": null,
    "catalog": "BASE",
    "policy": {
      "Version": "1.0",
      "Statement": [
        {
          "Action": [
            "::Get",
            "::List"
          ],
          "Effect": "Allow"
        },
        {
          "Action": [
            "identity:*"
          ],
          "Effect": "Deny"
        }
      ]
    },
    "id": "19bb93eec4ca4f08aefdc02da76d8f3c",
    "type": "AA",
    "name": "readonly"
  }
}
```

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.

4.8.3 Querying Permissions of a User Group Under a Domain

Function

This API is used to query the permissions of a user group under a domain. A role is a set of permissions and represents a group of actions.

URI

- URI format
GET /v3/domains/{domain_id}/groups/{group_id}/roles
- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID.
group_id	Yes	String	User group ID.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://sample.domain.com/v3/domains/d54061ebcb5145dd814f8eb3fe9b7ac0/groups/47d79cab2cf4c35b13493d919a5bb3d/roles
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
links	Yes	Dict	Role resource link of a specified user group under a domain.
roles	Yes	Array	Role of a specified user group under a domain.

- Role parameter description

Parameter	Mandatory	Type	Description
id	Yes	String	ID of a role of a specified user group under a domain.
links	Yes	Dict	Role resource link.
name	Yes	String	Name of a role.
domain_id	Yes	String	ID of the domain to which a role belongs.
type	Yes	String	Display mode of a role. <ul style="list-style-type: none"> AX: A role is displayed at the domain layer. XA: A role is displayed at the project layer. AA: A role is displayed at both the domain and project layers. XX: A role is not displayed at the domain or project layer.
display_name	No	String	Displayed name of a role.
catalog	No	String	Directory where a role locates.
policy	No	Dict	Policy of a role.
description	No	String	Description of a role.

- Example response

```
{
  "links": {
    "self": "https://sample.domain.com/v3/domains/d54061ebcb5145dd814f8eb3fe9b7ac0/groups/47d79cab2cf4c35b13493d919a5bb3d/roles",
    "previous": null,
    "next": null
  },
  "roles": [
    {
      "display_name": "Security Administrator",
```

```

    "description": "Security Administrator",
    "links": {
      "self": "https://sample.domain.com/v3/roles/005cf92cfd364105afaa5df2eec25012"
    },
    "domain_id": null,
    "name": "secu_admin",
    "type": "AX",
    "catalog": "BASE",
    "policy": {
      "Version": "1.0",
      "Statement": [
        {
          "Action": [
            "identity:*"
          ],
          "Effect": "Allow"
        }
      ]
    },
    "id": "005cf92cfd364105afaa5df2eec25012"
  },
  {
    "display_name": "Agent Operator",
    "description": "Agent Operator",
    "links": {
      "self": "https://sample.domain.com/v3/roles/d160d30477c642a486ad10e3b4d9820f"
    },
    "domain_id": null,
    "name": "te_agency",
    "type": "AX",
    "catalog": "IAM",
    "policy": {
      "Version": "1.0",
      "Statement": [
        {
          "Action": [
            "identity:assume role"
          ],
          "Effect": "Allow"
        }
      ]
    },
    "id": "d160d30477c642a486ad10e3b4d9820f"
  }
]

```

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.

4.8.4 Querying Permissions of a User Group Corresponding to a Project

Function

This API is used to query the permissions of a specified user group corresponding to a project. A role is a set of permissions and represents a group of actions.

URI

- URI format
GET /v3/projects/{project_id}/groups/{group_id}/roles
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.
group_id	Yes	String	ID of a user group.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://sample.domain.com/v3/projects/073bbf60da374853841cf6624c94de4b/groups/47d79cab2cf4c35b13493d919a5bb3d/roles
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
links	Yes	Dict	Role resource link.
roles	Yes	Array	List of roles.

- Role parameter description

Parameter	Mandatory	Type	Description
id	Yes	String	ID of a role.
links	Yes	Dict	Role resource link.

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a role.
domain_id	Yes	String	ID of the domain to which a role belongs.
type	Yes	String	Display mode of a role. <ul style="list-style-type: none"> ● AX: A role is displayed at the domain layer. ● XA: A role is displayed at the project layer. ● AA: A role is displayed at both the domain and project layers. ● XX: A role is not displayed at the domain or project layer.
display_name	No	String	Displayed name of a role.
catalog	No	String	Directory where a role locates.
policy	No	Dict	Policy of a role.
description	No	String	Description of a role.

- Example response

```
{
  "links": {
    "self": " https://sample.domain.com/v3/projects/3a4cd4d559d8492bbe7bd355643f9763/groups/728da352c017480f80b5a96beb15f0e6/roles",
    "previous": null,
    "next": null
  },
  "roles": [
    {
      "catalog": "BASE",
      "display_name": "Guest",
      "name": "readonly",
      "links": {
        "self": " https://sample.domain.com/v3/roles/13d132b7856945788f6df7eb3ed5c35e"
      },
      "policy": {
        "Version": "1.0",
        "Statement": [
          {
            "Action": [
              "**:Get**",
              "**:List**"
            ],
            "Effect": "Allow"
          },
          {
            "Action": [
              "identity:*"
            ],
            "Effect": "Deny"
          }
        ]
      }
    }
  ]
}
```



```

    },
    "domain_id": null,
    "type": "AA",
    "id": "13d132b7856945788f6df7eb3ed5c35e",
    "description": "Guest"
  },
  {
    "catalog": "BASE",
    "display_name": "Tenant Administrator",
    "name": "te_admin",
    "links": {
      "self": " https://sample.domain.com/v3/roles/1def304b73f14e8eb8d1eb9bf8337ae6"
    },
    "policy": {
      "Version": "1.0",
      "Statement": [
        {
          "Action": [
            "*"
          ],
          "Effect": "Allow"
        },
        {
          "Action": [
            "identity:*"
          ],
          "Effect": "Deny"
        }
      ]
    },
    "domain_id": null,
    "type": "AA",
    "id": "1def304b73f14e8eb8d1eb9bf8337ae6",
    "description": "Tenant Administrator"
  }
]

```

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.

4.8.5 Granting Permissions to a User Group of a Domain

Function

This API is used to grant permissions to a user group of a domain. A role is a set of permissions and represents a group of actions.

URI

- URI format
PUT /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}
- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the domain to which a user group belongs.
group_id	Yes	String	ID of a user group.
role_id	Yes	String	ID of a role.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X PUT https://sample.domain.com/v3/domains/d54061ebcb5145dd814f8eb3fe9b7ac0/groups/47d79cab2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2
```

Response Parameters

None

Status Codes

Status Code	Description
204	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.
409	A resource conflict occurs.

4.8.6 Granting Permissions to a User Group Corresponding to a Project

Function

This API is used to grant permissions to a user group corresponding to a project. A role is a set of permissions and represents a group of actions.

URI

- URI format
PUT /v3/projects/{project_id}/groups/{group_id}/roles/{role_id}
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.
group_id	Yes	String	ID of a user group.
role_id	Yes	String	ID of a role.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X PUT https://sample.domain.com/v3/projects/073bbf60da374853841cf6624c94de4b/groups/47d79cab2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2
```

Response Parameters

None

Status Codes

Status Code	Description
204	The request is successful.
400	The server failed to process the request.
401	Authentication failed.

Status Code	Description
403	Access denied.
404	The requested resource cannot be found.
409	A resource conflict occurs.

4.8.7 Deleting Permissions of a User Group Corresponding to a Project

Function

This API is used to delete permissions of a user group corresponding to a project. A role is a set of permissions and represents a group of actions.

URI

- URI format
DELETE /v3/projects/{project_id}/groups/{group_id}/roles/{role_id}
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.
group_id	Yes	String	ID of a user group.
role_id	Yes	String	ID of a role.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X DELETE https://sample.domain.com/v3/projects/073bbf60da374853841cf6624c94de4b/groups/47d79cab2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2
```

Response Parameters

None

Status Codes

Status Code	Description
204	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.

4.8.8 Deleting Permissions of a User Group of a Domain

Function

This API is used to delete permissions of a specified user group of a domain. A role is a set of permissions and represents a group of actions.

URI

- URI format
DELETE /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}
- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the domain to which a user group belongs.
group_id	Yes	String	ID of a user group.
role_id	Yes	String	ID of a role.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X DELETE https://sample.domain.com/v3/domains/d54061ebcb5145dd814f8eb3fe9b7ac0/groups/47d79cab2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2
```

Response Parameters

None

Status Codes

Status Code	Description
204	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.

4.8.9 Querying Whether a User Group Under a Domain Has Specific Permissions

Function

This API is used to query whether a specified user group under a domain has specific permissions. A role is a set of permissions and represents a group of actions.

URI

- URI format
HEAD /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}
- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the domain to which a user group belongs.
group_id	Yes	String	ID of a user group.
role_id	Yes	String	ID of a role.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X HEAD https://10.22.44.158:31943/v3/domains/d54061ebcb5145dd814f8eb3fe9b7ac0/groups/47d79cab2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2
```

Response Parameters

None

Status Codes

Status Code	Description
204	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.

4.8.10 Querying Whether a User Group Corresponding to a Project Has Specific Permissions

Function

This API is used to query whether a user group corresponding to a project has specific permissions. A role is a set of permissions and represents a group of actions.

URI

- URI format
HEAD /v3/projects/{project_id}/groups/{group_id}/roles/{role_id}
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.
group_id	Yes	String	ID of a user group.

Parameter	Mandatory	Type	Description
role_id	Yes	String	ID of a role.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X HEAD https://sample.domain.com/v3/projects/073bbf60da374853841cf6624c94de4b/groups/47d79cab2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2
```

Response Parameters

None

Status Codes

Status Code	Description
204	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.

4.8.11 Granting Permissions to a User Group for All Projects

Function

This API is used to grant permissions to a user group for all projects.

URI

- URI format
PUT /v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects

- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the domain to which a user group belongs.
group_id	Yes	String	ID of a user group.
role_id	Yes	String	User role ID.

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	User token (of a specified domain ID) with secu_admin permissions or a token with op_service or op_auth permissions.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -X PUT https://sample.domain.com/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects
```

Response Parameters

None

Status Codes

Status Code	Description
204	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.

4.8.12 Removing Specified Permissions of a User Group in All Projects

Function

This API is provided for the administrator to remove the specified permissions of a user group in all projects.

URI

DELETE /v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects

Table 4-81 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the domain to which the user group belongs.
group_id	Yes	String	User group ID.
role_id	Yes	String	Permission ID.

Request Parameters

Table 4-82 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-token	Yes	String	Token with Security Administrator or op_auth permissions.

Response Parameters

None

Example Request

```
DELETE https://sample.domain.com/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects
```

Example Response

None

Status Codes

Status Code	Description
204	The request is successful.
401	Authentication failed.
403	You do not have permission to perform this action.
404	The requested resource cannot be found.
500	Internal server error.

4.8.13 Checking Whether a User Group Has Specified Permissions for All Projects

Function

This API is provided for the to check whether a user group has specified permissions for all projects.

URI

HEAD /v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects

Table 4-83 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID. For details about how to obtain the ID, see 6.3 Obtaining User, Account, User Group, Project, and Agency Information .
group_id	Yes	String	User group ID. For details about how to obtain a user group ID, see 6.3 Obtaining User, Account, User Group, Project, and Agency Information .
role_id	Yes	String	Permission ID. For details about how to obtain a permission ID, see 4.8.1 Querying a Role List .

Request Parameters

Table 4-84 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

None

Example Request

```
HEAD https://v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects
```

Example Response

None

Status Codes

Status Code	Description
204	The request is successful.
401	Authentication failed.
403	Access denied.
404	The server could not find the requested page.

Error Codes

For details, see [Error Codes](#).

4.8.14 Querying All Permissions of a User Group

Function

This API is provided for the to query all permissions that have been assigned to a user group.

URI

```
GET /v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/  
inherited_to_projects
```

Table 4-85 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID. For details about how to obtain the ID, see 6.3 Obtaining User, Account, User Group, Project, and Agency Information .
group_id	Yes	String	User group ID. For details about how to obtain a user group ID, see 6.3 Obtaining User, Account, User Group, Project, and Agency Information .

Request Parameters

Table 4-86 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

Status code: 200

Table 4-87 Parameters in the response body

Parameter	Type	Description
links	object	Resource link information.
roles	Array of objects	Permission information.
total_number	Integer	Total number of custom policies. This parameter is returned only when domain_id is specified in the request.

Table 4-88 RoleResult

Parameter	Type	Description
domain_id	String	ID of the to which the permission belongs.

Parameter	Type	Description
flag	String	If this parameter is set to fine_grained , the permission is a system-defined policy.
description_cn	String	Description of the permission in Chinese.
catalog	String	Service catalog of the permission.
name	String	Permission name. This parameter is carried in the token of a user, allowing the system to determine whether the user has permissions to access a specific cloud service.
description	String	Description of the permission.
links	object	Permission resource link.
id	String	Permission ID.
display_name	String	Display name of the permission.
type	String	Display mode of the permission. NOTE <ul style="list-style-type: none"> • AX: Account level. • XA: Project level. • AA: Both the account level and project level. • XX: Neither the account level nor project level. • The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.
policy	object	Content of the permission.
updated_time	String	Time when the permission was last updated.
created_time	String	Time when the permission was created.

Table 4-89 Links

Parameter	Type	Description
self	String	Resource link.
previous	String	Previous resource link.
next	String	Next resource link.

Table 4-90 RolePolicy

Parameter	Type	Description
Depends	Array of objects	Dependent permissions.
Statement	Array of objects	Statement of the permission.
Version	String	Policy version. NOTE <ul style="list-style-type: none"> • 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. • 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.

Table 4-91 PolicyDepends

Parameter	Type	Description
catalog	String	Service catalog of the permission.
display_name	String	Display name of the permission.

Table 4-92 PolicyStatement

Parameter	Type	Description
Action	Array of strings	Specific operation permission on a resource. A maximum of 100 actions are allowed. NOTE <ul style="list-style-type: none"> • The value format is <i>Service name.Resource type.Operation</i>, for example, vpc:ports:create. • <i>Service name</i>: indicates the product name, such as ecs, evs, or vpc. Only lowercase letters are allowed. Resource types and operations are not case-sensitive. You can use an asterisk (*) to represent all operations. • In the case of a custom policy for agencies, this parameter should be set to "Action": <i>["iam:agencies:assume"]</i>.

Parameter	Type	Description
Effect	String	<p>Effect of the permission. The value can be Allow or Deny. If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements.</p> <p>Enumerated values:</p> <ul style="list-style-type: none"> • Allow • Deny
Condition	Object	<p>Conditions for the permission to take effect. A maximum of 10 conditions are allowed. For details about the condition parameters, see .</p> <p>NOTE Take the condition in the sample request as an example, the values of the condition key (obs:prefix) and string (public) must be equal (StringEquals).</p> <pre> "Condition": { "StringEquals": { "obs:prefix": ["public"] } } </pre>
Resource	Array of strings	<p>Cloud resource. The array can contain a maximum of 10 resource strings, and each string cannot exceed 128 characters.</p> <p>NOTE</p> <ul style="list-style-type: none"> • Format: <code>:::</code>. For example, obs::bucket:*. Asterisks are allowed. • The region segment can be <code>*</code> or a region accessible to the user. The specified resource must belong to the corresponding service that actually exists. • In the case of a custom policy for agencies, the type of this parameter is Object, and the value should be set to <code>"Resource": {"uri": ["/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"]}</code>.

Example Request

```
GET https://v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/inherited_to_projects
```

Example Response

Status code: 200

The request is successful.

```
{
  "roles": [ {
    "domain_id": null,
```



```

"description_cn" : "Description of the permission in Chinese",
"catalog" : "VulnScan",
"name" : "wscn_adm",
"description" : "Vulnerability Scan Service administrator of tasks and reports.",
"links" : {
  "next" : null,
  "previous" : null,
  "self" : "https://v3/roles/0af84c1502f447fa9c2fa18083fbb..."
},
"id" : "0af84c1502f447fa9c2fa18083fbb...",
"display_name" : "VSS Administrator",
"type" : "XA",
"policy" : {
  "Version" : "1.0",
  "Statement" : [ {
    "Action" : [ "WebScan:*:*" ],
    "Effect" : "Allow"
  } ],
  "Depends" : [ {
    "catalog" : "BASE",
    "display_name" : "Server Administrator"
  }, {
    "catalog" : "BASE",
    "display_name" : "Tenant Guest"
  } ]
}, {
  "domain_id" : null,
  "flag" : "fine_grained",
  "description_cn" : "Description of the permission in Chinese",
  "catalog" : "CSE",
  "name" : "system_all_34",
  "description" : "All permissions of CSE service.",
  "links" : {
    "next" : null,
    "previous" : null,
    "self" : "https://v3/roles/0b5ea44ebdc64a24a9c372b2317f7..."
  },
  "id" : "0b5ea44ebdc64a24a9c372b2317f7...",
  "display_name" : "CSE Admin",
  "type" : "XA",
  "policy" : {
    "Version" : "1.1",
    "Statement" : [ {
      "Action" : [ "cse:*:*", "ecs:*:*", "evs:*:*", "vpc:*:*" ],
      "Effect" : "Allow"
    } ]
  }
}, {
  "links" : {
    "next" : null,
    "previous" : null,
    "self" : "https://v3/roles"
  }
}

```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.

Error Codes

For details, see [Error Codes](#).

4.9 Custom Policy Management

4.9.1 Listing Custom Policies

Function

This API is provided for the administrator to list all custom policies.

The API can be called using both the global endpoint and region-specific endpoints.

URI

GET /v3.0/OS-ROLE/roles

Table 4-93 Query parameters

Parameter	Mandatory	Type	Description
page	No	Integer	Page number for pagination query. The minimum value is 1. This parameter must be used together with per_page .
per_page	No	Integer	Number of data records to be displayed on each page. The value ranges from 1 to 300. This parameter must be used together with page .

Request Parameters

Table 4-94 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	Token with Security Administrator permissions.

Response Parameters

Table 4-95 Parameters in the response body

Parameter	Type	Description
links	Object	Resource link information.
roles	Array of objects	Custom policy information.
total_number	Integer	Total number of custom policies returned.

Table 4-96 links

Parameter	Type	Description
self	String	Resource link.
previous	String	Previous resource link.
next	String	Next resource link.

Table 4-97 roles

Parameter	Type	Description
domain_id	String	ID of the account to which the custom policy belongs.
references	Integer	Number of references.
updated_time	String	Time when the custom policy was last updated.
created_time	String	Time when the custom policy was created.
description_cn	String	Description of the custom policy.
catalog	String	Service catalog.
name	String	Name of the custom policy.
description	String	Description of the custom policy.
links	Object	Resource link of the custom policy.

Parameter	Type	Description
id	String	Policy ID.
display_name	String	Display name of the custom policy.
type	String	Display mode. NOTE <ul style="list-style-type: none"> • AX: Account level. • XA: Project level. • The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.
policy	Object	Content of custom policy.

Table 4-98 roles.links

Parameter	Type	Description
self	String	Resource link.

Table 4-99 roles.policy

Parameter	Type	Description
Version	String	Policy version. NOTE <ul style="list-style-type: none"> • 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. • 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.
Statement	Array of objects	Statement of the policy. A policy can contain a maximum of eight statements.

Table 4-100 roles.policy.Statement

Parameter	Type	Description
Action	Array of strings	<p>Specific operation permission on a resource. A maximum of 100 actions are allowed.</p> <p>NOTE</p> <ul style="list-style-type: none"> The value format is <i>Service name.Resource type.Operation</i>, for example, vpc:ports:create. <i>Service name</i>: indicates the product name, such as ecs, evs, or vpc. Only lowercase letters are allowed. Resource types and operations are not case-sensitive. You can use an asterisk (*) to represent all operations. For a custom policy for agencies, this parameter should be set to <code>"Action": ["iam:agencies:assume"]</code>.
Effect	String	<p>Effect of the permission. The value can be Allow or Deny. If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements.</p>
Condition	Object	<p>Conditions for the permission to take effect. A maximum of 10 conditions are allowed.</p>
Resource	Array of strings	<p>Cloud resource. The array can contain a maximum of 10 resource strings, and each string cannot exceed 128 characters.</p> <p>NOTE</p> <ul style="list-style-type: none"> Format: <code>:::</code>. For example, obs::bucket:*. Asterisks are allowed. The region segment can be * or a region accessible to the user. The specified resource must belong to the corresponding service that actually exists. In the case of a custom policy for agencies, the type of this parameter is Object, and the value should be set to <code>"Resource": {"uri": ["iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"]}</code>.

Table 4-101 roles.policy.Statement.Condition

Parameter	Type	Description
operator	Object	<p>Operator, for example, Bool and StringEquals. The parameter type is custom object.</p>

Table 4-102 roles.policy.Statement.Condition.operator

Parameter	Type	Description
attribute	Array of strings	Condition key. The condition key must correspond to the specified operator. A maximum of 10 condition keys are allowed. The parameter type is custom character string array.

Example Request

GET <https://sample.domain.com/v3.0/OS-ROLE/roles>

Example Response

Status code: 200

The request is successful.

```
{
  "roles": [ {
    "domain_id": "d78cbac186b744899480f25bd022f...",
    "updated_time": "1579229246886",
    "created_time": "1579229246886",
    "description_cn": "Description in Chinese",
    "catalog": "CUSTOMED",
    "name": "custom_d78cbac186b744899480f25bd022f468_1",
    "description": "IAMDescription",
    "links": {
      "self": "https://sample.domain.com/v3/roles/93879fd90f1046f69e6e0b31c94d2..."
    },
  },
  {
    "id": "93879fd90f1046f69e6e0b31c94d2...",
    "display_name": "IAMCloudServicePolicy",
    "type": "AX",
    "policy": {
      "Version": "1.1",
      "Statement": [ {
        "Condition": {
          "StringStartWith": {
            "g:ProjectName": [ "AZ-1" ]
          }
        },
        "Action": [ "obs:bucket:GetBucketAcl" ],
        "Resource": [ "obs:*:bucket:*" ],
        "Effect": "Allow"
      } ]
    }
  },
  {
    "domain_id": "d78cbac186b744899480f25bd022f...",
    "updated_time": "1579229242358",
    "created_time": "1579229242358",
    "description_cn": "Description in Chinese",
    "catalog": "CUSTOMED",
    "name": "custom_d78cbac186b744899480f25bd022f468_0",
    "description": "IAMDescription",
    "links": {
      "self": "https://sample.domain.com/v3/roles/f67224e84dc849ab954ce29fb4f47..."
    },
  },
  {
    "id": "f67224e84dc849ab954ce29fb4f473...",
    "display_name": "IAMAgencyPolicy",
    "type": "AX",
    "policy": {
```

```

"Version" : "1.1",
"Statement" : [ {
  "Action" : [ "iam:agencies:assume" ],
  "Resource" : {
    "uri" : [ "iam/agencies/07805acaba800fdd4fbd00b8f888..." ]
  },
  "Effect" : "Allow"
} ]
} ],
"links" : {
  "next" : null,
  "previous" : null,
  "self" : "https://sample.domain.com/v3/roles?domain_id=d78cbac186b744899480f25bd022f..."
},
"total_number" : 300
}

```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
500	Internal server error.

Error Codes

None

4.9.2 Querying Custom Policy Details

Function

This API is provided for the administrator to query custom policy details.

The API can be called using both the global endpoint and region-specific endpoints.

URI

GET /v3.0/OS-ROLE/roles/{role_id}

Table 4-103 URI parameters

Parameter	Mandatory	Type	Description
role_id	Yes	String	Custom policy ID. For details about how to obtain a custom policy ID, see Custom Policy ID .

Request Parameters

Table 4-104 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	Token with Security Administrator permissions.

Response Parameters

Table 4-105 Parameters in the response body

Parameter	Type	Description
role	Object	Custom policy information.

Table 4-106 role

Parameter	Type	Description
domain_id	String	Domain ID.
references	Integer	Number of references.
updated_time	String	Time when the custom policy was last updated.
created_time	String	Time when the custom policy was created.
description_cn	String	Description of the custom policy.
catalog	String	Service catalog.
name	String	Name of the custom policy.
description	String	Description of the custom policy.

Parameter	Type	Description
links	Object	Resource link of the custom policy.
id	String	Policy ID.
display_name	String	Display name of the custom policy.
type	String	Display mode. NOTE <ul style="list-style-type: none"> • AX: Account level. • XA: Project level. • The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.
policy	Object	Content of custom policy.

Table 4-107 role.links

Parameter	Type	Description
self	String	Resource link.

Table 4-108 role.policy

Parameter	Type	Description
Version	String	Policy version. NOTE <ul style="list-style-type: none"> • 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. • 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.
Statement	Array of objects	Statement of the policy. A policy can contain a maximum of eight statements.

Table 4-109 role.policy.Statement

Parameter	Type	Description
Action	Array of strings	<p>Specific operation permission on a resource. A maximum of 100 actions are allowed.</p> <p>NOTE</p> <ul style="list-style-type: none"> The value format is <i>Service name.Resource type.Operation</i>, for example, vpc:ports:create. <i>Service name</i>: indicates the product name, such as ecs, evs, or vpc. Only lowercase letters are allowed. Resource types and operations are not case-sensitive. You can use an asterisk (*) to represent all operations. For a custom policy for agencies, this parameter should be set to "Action": <code>["iam:agencies:assume"]</code>.
Effect	String	<p>Effect of the permission. The value can be Allow or Deny. If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements.</p> <p>Options:</p> <ul style="list-style-type: none"> Allow Deny
Condition	Object	<p>Conditions for the permission to take effect. A maximum of 10 conditions are allowed.</p>
Resource	Array of strings	<p>Cloud resource. The array can contain a maximum of 10 resource strings, and each string cannot exceed 128 characters.</p> <p>NOTE</p> <ul style="list-style-type: none"> Format: <code>:::</code>. For example, obs::bucket:*. Asterisks are allowed. The region segment can be * or a region accessible to the user. The specified resource must belong to the corresponding service that actually exists. In the case of a custom policy for agencies, the type of this parameter is Object, and the value should be set to "Resource": <code>{"uri": ["iam/agencies/07805acaba800fdd4fbd00b8f888c7c"]}</code>.

Table 4-110 role.policy.Statement.Condition

Parameter	Type	Description
operator	Object	<p>Operator, for example, Bool and StringEquals.</p> <ul style="list-style-type: none"> The parameter type is custom object.

Table 4-111 role.policy.Statement.Condition.operator

Parameter	Type	Description
attribute	Array of strings	Condition key. The condition key must correspond to the specified operator. A maximum of 10 condition keys are allowed. <ul style="list-style-type: none"> The parameter type is custom character string array.

Example Request

GET https://sample.domain.com/v3.0/OS-ROLE/roles/{role_id}

Example Response

Status code: 200

The request is successful.

```
{
  "role": {
    "domain_id": "d78cbac186b744899480f25bd02...",
    "references": 0,
    "description_cn": "Policy description",
    "catalog": "CUSTOMED",
    "name": "custom_d78cbac186b744899480f25bd022f468_11",
    "description": "IAMDescription",
    "links": {
      "self": "https://sample.domain.com/v3/roles/a24a71dcc41f4da989c2a1c900b52d1a"
    },
    "id": "a24a71dcc41f4da989c2a1c900b52d1a",
    "display_name": "IAMCloudServicePolicy",
    "type": "AX",
    "policy": {
      "Version": "1.1",
      "Statement": [
        {
          "Condition": {
            "StringStartWith": {
              "g:ProjectName": [
                "example-west-1"
              ]
            }
          },
          "Action": [
            "obs:bucket:GetBucketAcl"
          ],
          "Resource": [
            "obs:*:bucket:*"
          ],
          "Effect": "Allow"
        }
      ]
    }
  }
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
500	Internal server error.

Error Codes

None

4.9.3 Creating a Custom Policy for Cloud Services

Function

This API is provided for the administrator to create a custom policy for cloud services.

The API can be called using both the global endpoint and region-specific endpoints.

URI

POST /v3.0/OS-ROLE/roles

Request Parameters

Table 4-112 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	Token with Security Administrator permissions.

Table 4-113 Parameters in the request body

Parameter	Mandatory	Type	Description
role	Yes	Object	Custom policy information.

Table 4-114 role

Parameter	Mandatory	Type	Description
display_name	Yes	String	Display name of the custom policy.
type	Yes	String	Display mode. NOTE <ul style="list-style-type: none"> • AX: Account level. • XA: Project level. • The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.
description	Yes	String	Description of the custom policy.
description_cn	No	String	Description of the custom policy.
policy	Yes	Object	Content of custom policy.

Table 4-115 role.policy

Parameter	Mandatory	Type	Description
Version	Yes	String	Policy version. When creating a custom policy, set this parameter to 1.1 . NOTE <ul style="list-style-type: none"> • 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. • 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.
Statement	Yes	Array of objects	Statement of the policy. A policy can contain a maximum of eight statements.

Table 4-116 role.policy.Statement

Parameter	Mandatory	Type	Description
Action	Yes	Array of strings	<p>Specific operation permission on a resource. A maximum of 100 actions are allowed.</p> <p>NOTE</p> <ul style="list-style-type: none"> The value format is <i>Service name.Resource type.Operation</i>, for example, vpc:ports:create. <i>Service name</i>: indicates the product name, such as ecs, evs, or vpc. Only lowercase letters are allowed. Resource types and operations are not case-sensitive. You can use an asterisk (*) to represent all operations.
Effect	Yes	String	<p>Effect of the permission. The value can be Allow or Deny. If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements.</p> <p>Options:</p> <ul style="list-style-type: none"> Allow Deny
Condition	No	Object	<p>Conditions for the permission to take effect. A maximum of 10 conditions are allowed.</p> <p>NOTE</p> <p>Take the condition in the sample request as an example, the condition key (obs:prefix) and the string (public) must be equal (StringEquals).</p> <pre> "Condition": { "StringEquals": { "obs:prefix": ["public"] } } </pre>

Parameter	Mandatory	Type	Description
Resource	No	Array of strings	<p>Cloud resource. The array can contain a maximum of 10 resource strings, and each string cannot exceed 128 characters.</p> <p>NOTE</p> <ul style="list-style-type: none"> Format: <code>.....</code>. For example, <code>obs::bucket:*</code>. Asterisks are allowed. The region segment can be <code>*</code> or a region accessible to the user. The specified resource must belong to the corresponding service that actually exists.

Response Parameters

Table 4-117 Parameters in the response body

Parameter	Type	Description
role	Object	Custom policy information.

Table 4-118 role

Parameter	Type	Description
catalog	String	Service catalog.
display_name	String	Display name of the custom policy.
description	String	Description of the custom policy.
links	Object	Resource link of the custom policy.
policy	Object	Content of custom policy.
description_cn	String	Description of the custom policy.
domain_id	String	Domain ID.
type	String	<p>Display mode.</p> <p>NOTE</p> <ul style="list-style-type: none"> AX: Account level. XA: Project level. The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.

Parameter	Type	Description
id	String	Policy ID.
name	String	Name of the custom policy.
updated_time	String	Time when the custom policy was last updated.
created_time	String	Time when the custom policy was created.
references	String	Number of references.

Table 4-119 role.links

Parameter	Type	Description
self	String	Resource link.

Table 4-120 role.policy

Parameter	Type	Description
Version	String	Policy version. NOTE <ul style="list-style-type: none"> • 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. • 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.
Statement	Array of objects	Statement of the policy. A policy can contain a maximum of eight statements.

Table 4-121 role.policy.Statement

Parameter	Type	Description
Action	Array of strings	Specific operation permission on a resource. A maximum of 100 actions are allowed. NOTE <ul style="list-style-type: none"> • The value format is <i>Service name.Resource type.Operation</i>, for example, vpc:ports:create. • <i>Service name</i>: indicates the product name, such as ecs, evs, or vpc. Only lowercase letters are allowed. Resource types and operations are not case-sensitive. You can use an asterisk (*) to represent all operations.

Parameter	Type	Description
Effect	String	<p>Effect of the permission. The value can be Allow or Deny. If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements.</p> <p>Options:</p> <ul style="list-style-type: none"> • Allow • Deny
Condition	Map<String,Map<String,Array<String>>>	<p>Conditions for the permission to take effect. A maximum of 10 conditions are allowed.</p> <p>NOTE Take the condition in the sample request as an example, the condition key (obs:prefix) and the string (public) must be equal (StringEquals).</p> <pre> "Condition": { "StringEquals": { "obs:prefix": ["public"] } } </pre>
Resource	Array of strings	<p>Cloud resource. The array can contain a maximum of 10 resource strings, and each string cannot exceed 128 characters.</p> <p>NOTE</p> <ul style="list-style-type: none"> • Format: <code>:::</code>. For example, obs:::bucket:*. Asterisks are allowed. • The region segment can be <code>*</code> or a region accessible to the user. The specified resource must belong to the corresponding service that actually exists.

Example Request

```

POST https://sample.domain.com/v3.0/OS-ROLE/roles
{
  "role": {
    "display_name": "IAMCloudServicePolicy",
    "type": "AX",
    "description": "IAMDescription",
    "description_cn": "Policy description",
    "policy": {
      "Version": "1.1",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "obs:bucket:GetBucketAcl"
          ],
          "Condition": {
            "StringStartWith": {
              "g:ProjectName": [
                "example-west-1"
              ]
            }
          }
        }
      ]
    }
  }
}

```

```
}  
  },  
  ],  
}  
}
```

Example Response

Status code: 201

The request is successful.

```
{  
  "role": {  
    "catalog": "CUSTOMED",  
    "display_name": "IAMCloudServicePolicy",  
    "description": "IAMDescription",  
    "links": {  
      "self": "https://sample.domain.com/v3/roles/93879fd90f1046f69e6e0b31c94d2..."  
    },  
    "policy": {  
      "Version": "1.1",  
      "Statement": [  
        {  
          "Action": [  
            "obs:bucket:GetBucketAcl"  
          ],  
          "Resource": [  
            "obs:*:bucket:*"  
          ],  
          "Effect": "Allow",  
          "Condition": {  
            "StringStartWith": {  
              "g:ProjectName": [  
                "example-west-1"  
              ]  
            }  
          }  
        ]  
      ]  
    }  
  },  
  "description_cn": "Policy description",  
  "domain_id": "d78cbac186b744899480f25bd...",  
  "type": "AX",  
  "id": "93879fd90f1046f69e6e0b31c9...",  
  "name": "custom_d78cbac186b744899480f25bd022f468_1"  
}
```

Status Codes

Status Code	Description
201	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
500	Internal server error.

Error Codes

None

4.9.4 Creating a Custom Policy for Agencies

Function

This API is provided for the administrator to create a custom policy for agencies.

The API can be called using both the global endpoint and region-specific endpoints.

URI

POST /v3.0/OS-ROLE/roles

Request Parameters

Table 4-122 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	Token with Security Administrator permissions.

Table 4-123 Parameters in the request body

Parameter	Mandatory	Type	Description
role	Yes	Object	Custom policy information.

Table 4-124 role

Parameter	Mandatory	Type	Description
display_name	Yes	String	Display name of the custom policy.

Parameter	Mandatory	Type	Description
type	Yes	String	Display mode. NOTE <ul style="list-style-type: none"> • AX: Account level. • XA: Project level. • The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.
description	Yes	String	Description of the custom policy.
description_cn	No	String	Description of the custom policy.
policy	Yes	Object	Content of custom policy.

Table 4-125 role.policy

Parameter	Mandatory	Type	Description
Version	Yes	String	Policy version. When creating a custom policy, set this parameter to 1.1 . NOTE <ul style="list-style-type: none"> • 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. • 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.
Statement	Yes	Array of objects	Statement of the policy. A policy can contain a maximum of eight statements.

Table 4-126 role.policy.Statement

Parameter	Mandatory	Type	Description
Action	Yes	Array of strings	<p>An action item is a specific operation permission on a resource.</p> <p>NOTE</p> <ul style="list-style-type: none"> For a custom policy for agencies, this parameter should be set to "Action": ["iam:agencies:assume"]. <p>Options:</p> <ul style="list-style-type: none"> iam:agencies:assume
Effect	Yes	String	<p>Effect of the permission. The value can be Allow or Deny. If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements.</p> <p>Options:</p> <ul style="list-style-type: none"> Allow Deny
Resource	No	Object	<p>Resources to be managed. After an account establishes multiple trust relationships between itself and your account, you can authorize IAM users in different user groups to manage resources of the delegating party. Each IAM user can only switch to the delegated agencies. For example:</p> <pre>"Resource": {"uri": ["/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"]}</pre>

Table 4-127 role.policy.Statement.Resource

Parameter	Mandatory	Type	Description
uri	Yes	Array of strings	<p>URI of a delegated resource, which can contain a maximum of 128 characters. Format: /iam/agencies/delegation ID. For example:</p> <pre>"uri": ["/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"]</pre>

Response Parameters

Table 4-128 Parameters in the response body

Parameter	Type	Description
role	Object	Custom policy information.

Table 4-129 role

Parameter	Type	Description
catalog	String	Service catalog.
display_name	String	Display name of the custom policy.
description	String	Description of the custom policy.
links	Object	Resource link of the custom policy.
policy	Object	Content of custom policy.
description_cn	String	Description of the custom policy.
domain_id	String	Domain ID.
type	String	Display mode. NOTE <ul style="list-style-type: none"> • AX: Account level. • XA: Project level. • The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.
id	String	Policy ID.
name	String	Name of the custom policy.
updated_time	String	Time when the custom policy was last updated.
created_time	String	Time when the custom policy was created.
references	String	Number of references.

Table 4-130 role.links

Parameter	Type	Description
self	String	Resource link.

Table 4-131 role.policy

Parameter	Type	Description
Version	String	Policy version. NOTE <ul style="list-style-type: none"> • 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. • 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.
Statement	Array of objects	Statement of the policy. A policy can contain a maximum of eight statements.

Table 4-132 role.policy.Statement

Parameter	Type	Description
Action	Array of strings	An action item is a specific operation permission on a resource. NOTE <ul style="list-style-type: none"> • For a custom policy for agencies, this parameter should be set to <i>"Action": ["iam:agencies:assume"]</i>.
Effect	String	Effect of the permission. The value can be Allow or Deny . If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements. Options: <ul style="list-style-type: none"> • Allow • Deny
Resource	Object	Resources to be managed. After an account establishes multiple trust relationships between itself and your account, you can authorize IAM users in different user groups to manage resources of the delegating party. Each IAM user can only switch to the delegated agencies. For example: "Resource": {"uri": ["/iam/agencies/07805acaba800fdd4fbd00b8f888c7c"]}

Table 4-133 role.policy.Statement.Resource

Parameter	Type	Description
uri	Array of strings	URI of a delegated resource, which can contain a maximum of 128 characters. Format: /iam/agencies/ <i>delegation ID</i> . For example: "uri": ["/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"]

Example Request

```
POST https://sample.domain.com/v3.0/OS-ROLE/roles
{
  "role": {
    "display_name": "IAMAgencyPolicy",
    "type": "AX",
    "description": "IAMDescription",
    "description_cn": "Policy description",
    "policy": {
      "Version": "1.1",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "iam:agencies:assume"
          ],
          "Resource": {
            "uri": [
              "/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"
            ]
          }
        }
      ]
    }
  }
}
```

Example Response

Status code: 201

The request is successful.

```
{
  "role": {
    "catalog": "CUSTOMED",
    "display_name": "IAMAgencyPolicy",
    "description": "IAMDescription",
    "links": {
      "self": "https://sample.domain.com/v3/roles/f67224e84dc849ab954ce29fb4f47f8e"
    },
    "policy": {
      "Version": "1.1",
      "Statement": [
        {
          "Action": [
            "iam:agencies:assume"
          ],
          "Resource": {
            "uri": [
              "/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"
            ]
          },
          "Effect": "Allow"
        }
      ]
    }
  }
}
```



```

    }
  ]
},
"description_cn": "Policy description",
"domain_id": "d78cbac186b744899480f25bd02...",
"type": "AX",
"id": "f67224e84dc849ab954ce29fb4f47f8e",
"name": "custom_d78cbac186b744899480f25bd022f468_0"
}
}

```

Status Codes

Status Code	Description
201	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
500	Internal server error.

Error Codes

None

4.9.5 Modifying a Custom Policy for Cloud Services

Function

This API is provided for the administrator to modify a custom policy for cloud services.

The API can be called using both the global endpoint and region-specific endpoints.

URI

PATCH /v3.0/OS-ROLE/roles/{role_id}

Table 4-134 URI parameters

Parameter	Mandatory	Type	Description
role_id	Yes	String	Custom policy ID. For details about how to obtain a custom policy ID, see Custom Policy ID .

Request Parameters

Table 4-135 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	Token with Security Administrator permissions.

Table 4-136 Parameters in the request body

Parameter	Mandatory	Type	Description
role	Yes	Object	Custom policy information.

Table 4-137 role

Parameter	Mandatory	Type	Description
display_name	Yes	String	Display name of the custom policy.
type	Yes	String	Display mode. NOTE <ul style="list-style-type: none"> • AX: Account level. • XA: Project level. • The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.
description	Yes	String	Description of the custom policy.
description_cn	No	String	Description of the custom policy.
policy	Yes	Object	Content of custom policy.

Table 4-138 role.policy

Parameter	Mandatory	Type	Description
Version	Yes	String	Policy version. When creating a custom policy, set this parameter to 1.1 . NOTE <ul style="list-style-type: none"> 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.
Statement	Yes	Array of objects	Statement of the policy. A policy can contain a maximum of eight statements.

Table 4-139 role.policy.Statement

Parameter	Mandatory	Type	Description
Action	Yes	Array of strings	Specific operation permission on a resource. A maximum of 100 actions are allowed. NOTE <ul style="list-style-type: none"> The value format is <i>Service name.Resource type.Operation</i>, for example, vpc:ports:create. <i>Service name</i>: indicates the product name, such as ecs, evs, or vpc. Only lowercase letters are allowed. Resource types and operations are not case-sensitive. You can use an asterisk (*) to represent all operations.
Effect	Yes	String	Effect of the permission. The value can be Allow or Deny . If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements. Options: <ul style="list-style-type: none"> Allow Deny

Parameter	Mandatory	Type	Description
Condition	No	Object	Conditions for the permission to take effect. A maximum of 10 conditions are allowed.
Resource	No	Array of strings	Cloud resource. The array can contain a maximum of 10 resource strings, and each string cannot exceed 128 characters. NOTE <ul style="list-style-type: none"> Format: <code>.....</code>. For example, <code>obs::bucket:*</code>. Asterisks are allowed. The region segment can be <code>*</code> or a region accessible to the user. The specified resource must belong to the corresponding service that actually exists.

Table 4-140 role.policy.Statement.Condition

Parameter	Mandatory	Type	Description
<operator>	No	Object	Operator, for example, Bool and StringEquals. <ul style="list-style-type: none"> The parameter type is custom object.

Table 4-141 role.policy.Statement.Condition.<operator>

Parameter	Mandatory	Type	Description
<attribute>	No	Array of strings	Condition key. The condition key must correspond to the specified operator. A maximum of 10 condition keys are allowed. <ul style="list-style-type: none"> The parameter type is custom character string array.

Response Parameters

Table 4-142 Parameters in the response body

Parameter	Type	Description
role	Object	Custom policy information.

Table 4-143 role

Parameter	Type	Description
catalog	String	Service catalog.
display_name	String	Display name of the custom policy.
description	String	Description of the custom policy.
links	Object	Resource link of the custom policy.
policy	Object	Content of custom policy.
description_cn	String	Description of the custom policy.
domain_id	String	Domain ID.
type	String	Display mode. NOTE <ul style="list-style-type: none"> • AX: Account level. • XA: Project level. • The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.
id	String	Policy ID.
name	String	Name of the custom policy.
updated_time	String	Time when the custom policy was last updated.
created_time	String	Time when the custom policy was created.
references	String	Number of references.

Table 4-144 role.links

Parameter	Type	Description
self	String	Resource link.

Table 4-145 role.policy

Parameter	Type	Description
Version	String	Policy version. NOTE <ul style="list-style-type: none"> • 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. • 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.
Statement	Array of objects	Statement of the policy. A policy can contain a maximum of eight statements.

Table 4-146 role.policy.Statement

Parameter	Type	Description
Action	Array of strings	Specific operation permission on a resource. A maximum of 100 actions are allowed. NOTE <ul style="list-style-type: none"> • The value format is <i>Service name.Resource type.Operation</i>, for example, vpc:ports:create. • <i>Service name</i>: indicates the product name, such as ecs, evs, or vpc. Only lowercase letters are allowed. Resource types and operations are not case-sensitive. You can use an asterisk (*) to represent all operations.
Effect	String	Effect of the permission. The value can be Allow or Deny . If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements. Options: <ul style="list-style-type: none"> • Allow • Deny
Condition	Object	Conditions for the permission to take effect. A maximum of 10 conditions are allowed.

Parameter	Type	Description
Resource	Array of strings	<p>Cloud resource. The array can contain a maximum of 10 resource strings, and each string cannot exceed 128 characters.</p> <p>NOTE</p> <ul style="list-style-type: none"> • Format: <code>:::</code>. For example, <code>obs::bucket:*</code>. Asterisks are allowed. • The region segment can be <code>*</code> or a region accessible to the user. The specified resource must belong to the corresponding service that actually exists.

Table 4-147 role.policy.Statement.Condition

Parameter	Type	Description
<code>operator</code>	Object	<p>Operator, for example, Bool and StringEquals.</p> <ul style="list-style-type: none"> • The parameter type is custom object.

Table 4-148 role.policy.Statement.Condition.operator

Parameter	Type	Description
attribute	Array of strings	<p>Condition key. The condition key must correspond to the specified operator. A maximum of 10 condition keys are allowed.</p> <ul style="list-style-type: none"> • The parameter type is custom character string array.

Example Request

```
PATCH https://sample.domain.com/v3.0/OS-ROLE/roles/{role_id}
{
  "role": {
    "display_name": "IAMCloudServicePolicy",
    "type": "AX",
    "description": "IAMDescription",
    "description_cn": "Policy description",
    "policy": {
      "Version": "1.1",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "obs:bucket:GetBucketAcl"
          ],
          "Condition": {
            "StringStartWith": {
              "g:ProjectName": [
                "example-west-1"
              ]
            }
          }
        }
      ]
    }
  }
}
```

```
    },  
    "Resource": [  
      "obs:*:bucket:*"  
    ]  
  }  
]  
}  
}
```

Example Response

Status code: 200

The request is successful.

```
{  
  "role": {  
    "catalog": "CUSTOMED",  
    "display_name": "IAMCloudServicePolicy",  
    "description": "IAMDescription",  
    "links": {  
      "self": "https://sample.domain.com/v3/roles/93879fd90f1046f69e6e0b31c94d2615"  
    },  
    "policy": {  
      "Version": "1.1",  
      "Statement": [  
        {  
          "Action": [  
            "obs:bucket:GetBucketAcl"  
          ],  
          "Resource": [  
            "obs:*:bucket:*"  
          ],  
          "Effect": "Allow",  
          "Condition": {  
            "StringStartWith": {  
              "g:ProjectName": [  
                "example-west-1"  
              ]  
            }  
          }  
        }  
      ]  
    },  
    "description_cn": "Policy description",  
    "domain_id": "d78cbac186b744899480f25bd0...",  
    "type": "AX",  
    "id": "93879fd90f1046f69e6e0b31c94d2615",  
    "name": "custom_d78cbac186b744899480f25bd022f468_1"  
  }  
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.

Status Code	Description
404	The requested resource cannot be found.
500	Internal server error.

Error Codes

None

4.9.6 Modifying a Custom Policy for Agencies

Function

This API is provided for the administrator to modify a custom policy for agencies.

The API can be called using both the global endpoint and region-specific endpoints.

URI

PATCH /v3.0/OS-ROLE/roles/{role_id}

Table 4-149 URI parameters

Parameter	Mandatory	Type	Description
role_id	Yes	String	Custom policy ID. For details about how to obtain a custom policy ID, see Custom Policy ID .

Request Parameters

Table 4-150 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	Token with Security Administrator permissions.

Table 4-151 Parameters in the request body

Parameter	Mandatory	Type	Description
role	Yes	Object	Custom policy information.

Table 4-152 role

Parameter	Mandatory	Type	Description
display_name	Yes	String	Display name of the custom policy.
type	Yes	String	Display mode. NOTE <ul style="list-style-type: none"> • AX: Account level. • XA: Project level. • The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.
description	Yes	String	Description of the custom policy.
description_cn	No	String	Description of the custom policy.
policy	Yes	Object	Content of custom policy.

Table 4-153 role.policy

Parameter	Mandatory	Type	Description
Version	Yes	String	Policy version. When creating a custom policy, set this parameter to 1.1 . NOTE <ul style="list-style-type: none"> • 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. • 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.
Statement	Yes	Array of objects	Statement of the policy. A policy can contain a maximum of eight statements.

Table 4-154 role.policy.Statement

Parameter	Mandatory	Type	Description
Action	Yes	Array of strings	<p>An action item is a specific operation permission on a resource.</p> <p>NOTE</p> <ul style="list-style-type: none"> For a custom policy for agencies, this parameter should be set to "Action": ["iam:agencies:assume"]. <p>Options:</p> <ul style="list-style-type: none"> iam:agencies:assume
Effect	Yes	String	<p>Effect of the permission. The value can be Allow or Deny. If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements.</p> <p>Options:</p> <ul style="list-style-type: none"> Allow Deny
Resource	No	Object	<p>Resources to be managed. After an account establishes multiple trust relationships between itself and your account, you can authorize IAM users in different user groups to manage resources of the delegating party. Each IAM user can only switch to the delegated agencies. For example:</p> <pre>"Resource": {"uri": ["/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"]}</pre>

Table 4-155 role.policy.Statement.Resource

Parameter	Mandatory	Type	Description
uri	Yes	Array of strings	<p>URI of a delegated resource, which can contain a maximum of 128 characters. Format: /iam/agencies/delegation ID. For example:</p> <pre>"uri": ["/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"]</pre>

Response Parameters

Table 4-156 Parameters in the response body

Parameter	Type	Description
role	Object	Custom policy information.

Table 4-157 role

Parameter	Type	Description
catalog	String	Service catalog.
display_name	String	Display name of the custom policy.
description	String	Description of the custom policy.
links	Object	Resource link of the custom policy.
policy	Object	Content of custom policy.
description_cn	String	Description of the custom policy.
domain_id	String	Domain ID.
type	String	Display mode. NOTE <ul style="list-style-type: none"> • AX: Account level. • XA: Project level. • The display mode of a custom policy can only be AX or XA. A custom policy must be displayed at either of the two levels.
id	String	Policy ID.
name	String	Name of the custom policy.
updated_time	String	Time when the custom policy was last updated.
created_time	String	Time when the custom policy was created.
references	String	Number of references.

Table 4-158 role.links

Parameter	Type	Description
self	String	Resource link.

Table 4-159 role.policy

Parameter	Type	Description
Version	String	Policy version. NOTE <ul style="list-style-type: none"> • 1.0: System-defined role. Only a limited number of service-level roles are provided for authorization. • 1.1: Policy. A policy defines the permissions required to perform operations on a specific cloud resource under certain conditions.
Statement	Array of objects	Statement of the policy. A policy can contain a maximum of eight statements.

Table 4-160 role.policy.Statement

Parameter	Type	Description
Action	Array of strings	An action item is a specific operation permission on a resource. NOTE <ul style="list-style-type: none"> • For a custom policy for agencies, this parameter should be set to <i>"Action": ["iam:agencies:assume"]</i>.
Effect	String	Effect of the permission. The value can be Allow or Deny . If both Allow and Deny statements are found in a policy, the authentication starts from the Deny statements. Options: <ul style="list-style-type: none"> • Allow • Deny
Resource	Object	Resources to be managed. After an account establishes multiple trust relationships between itself and your account, you can authorize IAM users in different user groups to manage resources of the delegating party. Each IAM user can only switch to the delegated agencies. For example: "Resource": {"uri": ["/iam/agencies/07805acaba800fdd4fbd00b8f888c7c"]}

Table 4-161 role.policy.Statement.Resource

Parameter	Type	Description
uri	Array of strings	URI of a delegated resource, which can contain a maximum of 128 characters. Format: /iam/agencies/ <i>delegation ID</i> . For example: "uri": ["/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"]

Example Request

```
PATCH https://sample.domain.com/v3.0/OS-ROLE/roles/{role_id}
{
  "role": {
    "display_name": "IAMAgencyPolicy",
    "type": "AX",
    "description": "IAMDescription",
    "description_cn": "Policy description",
    "policy": {
      "Version": "1.1",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "iam:agencies:assume"
          ],
          "Resource": {
            "uri": [
              "/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"
            ]
          }
        }
      ]
    }
  }
}
```

Example Response

Status code: 200

The request is successful.

```
{
  "role": {
    "catalog": "CUSTOMED",
    "display_name": "IAMAgencyPolicy",
    "description": "IAMDescription",
    "links": {
      "self": "https://sample.domain.com/v3/roles/f67224e84dc849ab954ce29fb4f47f8e"
    },
    "policy": {
      "Version": "1.1",
      "Statement": [
        {
          "Action": [
            "iam:agencies:assume"
          ],
          "Resource": {
            "uri": [
              "/iam/agencies/07805acaba800fdd4fbdc00b8f888c7c"
            ]
          },
          "Effect": "Allow"
        }
      ]
    }
  }
}
```

```

    }
  ]
},
"description_cn": "Policy description",
"domain_id": "d78cbac186b744899480f25b...",
"type": "AX",
"id": "f67224e84dc849ab954ce29fb4f47f8e",
"name": "custom_d78cbac186b744899480f25bd022f468_0"
}
}

```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
500	Internal server error.

Error Codes

None

4.9.7 Deleting a Custom Policy

Function

This API is provided for the administrator to delete a custom policy.

The API can be called using both the global endpoint and region-specific endpoints.

URI

DELETE /v3.0/OS-ROLE/roles/{role_id}

Table 4-162 URI parameters

Parameter	Mandatory	Type	Description
role_id	Yes	String	Custom policy ID. For details about how to obtain a custom policy ID, see Custom Policy ID .

Request Parameters

Table 4-163 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	Token with Security Administrator permissions.

Response Parameters

None

Example Request

```
DELETE https://sample.domain.com/v3.0/OS-ROLE/roles/{role_id}
```

Example Response

None

Status Codes

Status Code	Description
200	The custom policy is deleted successfully.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
500	Internal server error.

Error Codes

None

4.10 Agency Management

4.10.1 Creating an Agency

Function

This API is used to create an agency.

URI

POST /v3.0/OS-AGENCY/agencies

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	application/json;charset=utf8
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
name	Yes	String	Name of an agency. The length is less than or equal to 64 characters.
domain_id	Yes	String	ID of the current domain.
trust_domain_id	At least one	String	ID of the delegated domain.
trust_domain_name		String	Name of the delegated domain.
description	No	String	Description of an agency. The length is less than or equal to 255 characters.
duration	No	String	Validity period of the agency. The default value is null , which means that the agency will never expire. If this parameter is set to FOREVER , the validity of the agency is unlimited. If it is set to ONEDAY , the agency is valid only for one day.

NOTE

At least one of **trust_domain_id** and **trust_domain_name** must exist in the request body. If both of them exist, **trust_domain_name** takes precedence.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X POST -d '{"agency": {"name": "exampleagency", "domain_id": "0ae9c6993a2e47bb8c4c7a9bb8278d61", "trust_domain_id":
```

```
"35d7706cedbc49a18df0783d00269c20","trust_domain_name": "exampledomain","description": "testsfdas"}}' https://v3.0/OS-AGENCY/agencies
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
agency	Yes	JSON object	Delegated object.

- Description for the agency format

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an agency.
name	Yes	String	Name of an agency.
domain_id	Yes	String	ID of the current domain.
trust_domain_id	Yes	String	ID of the delegated domain.
description	Yes	String	Description of an agency.
duration	Yes	String	Validity period of an agency.
expire_time	Yes	String	Expiration time of an agency.
create_time	Yes	String	Time when an agency is created.

- Example response (request successful)

```
{
  "agency": {
    "description": "testsfdas",
    "trust_domain_id": "35d7706cedbc49a18df0783d00269c20",
    "id": "c1a06ec7387f430c8122d6f336c66dcf",
    "duration": null,
    "create_time": "2017-01-06T05:56:09.738212",
    "expire_time": null,
    "domain_id": "0ae9c6993a2e47bb8c4c7a9bb8278d61",
    "name": "exampleagency"
  }
}
```

- Example response (request failed)

```
{
  "error": {
    "message": "'name' is a required property",
    "code": 400,
    "title": "Bad Request"
  }
}
```

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.
409	The agency already exists.
500	Internal server error.

4.10.2 Querying an Agency List Based on the Specified Conditions

Function

This API is used to query an agency list based on the specified conditions.

URI

- URI format
GET /v3.0/OS-AGENCY/agencies{?domain_id,name,trust_domain_id}
- Query parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the current domain.
name	No	String	Name of an agency.
trust_domain_id	No	String	ID of the delegated domain.

Request Parameters

- Parameters in the request header

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

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://v3.0/OS-AGENCY/agencies?domain_id=0ae9c6993a2e47bb8c4c7a9bb8278d61
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
agencies	Yes	JSONArray	List of agencies.

- Description for the agency format

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an agency.
name	Yes	String	Name of an agency.
domain_id	Yes	String	ID of the current domain.
trust_domain_id	Yes	String	ID of the delegated domain.
trust_domain_name	Yes	String	Name of the delegated domain.
description	Yes	String	Description of an agency.
duration	Yes	String	Validity period of an agency. The default value is null , indicating that the agency is permanently valid.
expire_time	Yes	String	Expiration time of an agency.
create_time	Yes	String	Time when an agency is created.

- Example response (request successful)

```
{
  "agencies": [
    {
      "trust_domain_name": "exampdomain",
      "description": " testsfdas ",
      "trust_domain_id": "b3f266d0c08544a0859740de8b84e850",
      "id": "afca8ddf2e92469a8fd26a635da5206f",
      "duration": null,
      "create_time": "2017-01-04T09:09:15.000000",
      "expire_time": null,
    }
  ]
}
```

```

    "domain_id": "0ae9c6993a2e47bb8c4c7a9bb8278d61",
    "name": "exampleagency"
  }
]
}

```

- Example response (request failed)

```

{
  "error": {
    "message": "You are not authorized to perform the requested action: identity:list_agencies",
    "code": 403,
    "title": "Forbidden"
  }
}

```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.10.3 Obtaining Details of a Specified Agency

Function

This API is used to obtain the details of a specified agency.

URI

- URI format
GET /v3.0/OS-AGENCY/agencies/{agency_id}
- URI parameters

Parameter	Mandatory	Type	Description
agency_id	Yes	String	ID of an agency.

Request Parameters

- Parameters in the request header

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

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://v3.0/OS-AGENCY/agencies/2809756f748a46e2b92d58d309f67291
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
agency	Yes	JSON object	Delegated object.

- Description for the agency format

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an agency.
name	Yes	String	Name of an agency.
domain_id	Yes	String	ID of the current domain.
trust_domain_id	Yes	String	ID of the delegated domain.
trust_domain_name	Yes	String	Name of the delegated domain.
description	Yes	String	Description of an agency.
duration	Yes	String	Validity period of an agency. The default value is null , indicating that the agency is permanently valid.
expire_time	Yes	String	Expiration time of an agency.
create_time	Yes	String	Time when an agency is created.

- Example response (request successful)

```
{
  "agency": {
    "description": " testsfdas ",
    "trust_domain_id": "3ebe1024db46485cb02ef08d3c348477",
    "trust_domain_name": "exampledomain",
    "id": "c1a06ec7387f430c8122d6f336c66dcf",
    "duration": "FOREVER",
    "create_time": "2017-01-06T05:56:09.738212",
    "expire_time": null,
    "domain_id": "0ae9c6993a2e47bb8c4c7a9bb8278d61",
  }
}
```

```
"name" : "exampleagency"
}
```

- Example response (request failed)

```
{
  "error": {
    "message": "Could not find agency: 2809756f748a46e2b92d58d309f67291",
    "code": 404,
    "title": "Not Found"
  }
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The agency does not exist.
500	Internal server error.

4.10.4 Modifying an Agency

Function

This API is used to modify agency information, including the **trust_domain_id**, **description**, and **trust_domain_name** parameters.

URI

- URI format
PUT /v3.0/OS-AGENCY/agencies/{agency_id}
- URI parameters

Parameter	Mandatory	Type	Description
agency_id	Yes	String	ID of an agency.

Request Parameters

- Parameters in the request header

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

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
trust_domain_id	No	String	ID of the delegated domain. The delegated domain must exist.
trust_domain_name	No	String	Name of the delegated domain. The delegated domain must exist.
description	No	String	Description of an agency.

 **NOTE**

The **trust_domain_id** and **trust_domain_name** parameters in a request body must exist or not exist at the same time. If both of them exist, **trust_domain_name** takes precedence.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X PUT -d '{"agency": {"trust_domain_id": "35d7706cedbc49a18df0783d00269c20", "trust_domain_name": "exampledomain", "description": "111111"}}' https://v3.0/OS-AGENCY/agencies/2809756f748a46e2b92d58d309f67291
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
agency	Yes	JSON object	Delegated object.

- Description for the agency format

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an agency.
name	Yes	String	Name of an agency.
domain_id	Yes	String	ID of the current domain.

Parameter	Mandatory	Type	Description
trust_domain_id	Yes	String	ID of the delegated domain.
description	Yes	String	Description of an agency.
duration	Yes	String	Validity period of an agency. The default value is null , indicating that the agency is permanently valid.
expire_time	Yes	String	Expiration time of an agency.
create_time	Yes	String	Time when an agency is created.

- Example response (request successful)

```
{
  "agency": {
    "description": " testsfdas ",
    "trust_domain_id": "3ebe1024db46485cb02ef08d3c348477",
    "id": "c1a06ec7387f430c8122d6f336c66dcf",
    "duration": null,
    "create_time": "2017-01-06T05:56:09.738212",
    "expire_time": null,
    "domain_id": "0ae9c6993a2e47bb8c4c7a9bb8278d61",
    "name": "exampleagency"
  }
}
```

- Example response (request failed)

```
{
  "error": {
    "message": "TrustDomainNotFound",
    "code": 404,
    "title": "Not Found"
  }
}
```

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.

4.10.5 Deleting an Agency

Function

This API is used to delete an agency.

NOTE

After this operation, the delegated party can no longer access the relevant resources. Exercise caution when performing this operation.

URI

- URI format
DELETE /v3.0/OS-AGENCY/agencies/{agency_id}
- URI parameters

Parameter	Mandatory	Type	Description
agency_id	Yes	String	ID of an agency.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X DELETE https://v3.0/OS-AGENCY/agencies/2809756f748a46e2b92d58d309f67291
```

Response Parameters

- Example response (request failed)

```
{
  "error": {
    "message": "Could not find agency: 2809756f748a46e2b92d58d309f67291",
    "code": 404,
    "title": "Not Found"
  }
}
```

Status Codes

Status Code	Description
204	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.10.6 Granting Permissions to an Agency for a Project

Function

This API is used to grant permissions to an agency for a project.

URI

- URI format
PUT /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id}
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	ID of a project under the current domain.
agency_id	Yes	String	ID of an agency.
role_id	Yes	String	ID of a role.

NOTE

The role name corresponding to **role_id** in a request body is controlled by a blacklist. The role name cannot be **secu_admin** or **te_agency**.

Request Parameters

- Parameters in the request header

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

Parameter	Mandator y	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X PUT https://v3.0/OS-AGENCY/projects/0945241c5ebc4660bac540d48f2a2c14/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff
```

Response Parameters

- No response: indicates that the response is successful.
- Example response (request failed)

```
{
  "error": {
    "message": "Could not find role: 0f3a2d418ed747fa8be46e92757be9ddff",
    "code": 404,
    "title": "Not Found"
  }
}
```

Status Codes

Status Code	Description
204	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.10.7 Checking Whether an Agency Has the Specified Permissions on a Project

Function

This API is used to check whether an agency has the specified permissions on a project.

URI

- URI format
HEAD /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id}

- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	ID of a project under the current domain.
agency_id	Yes	String	ID of an agency.
role_id	Yes	String	ID of a role.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X HEAD https://v3.0/OS-AGENCY/projects/0945241c5ebc4660bac540d48f2a2c14/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff
```

Response Parameters

- Example response (request failed)

```
{
  "error": {
    "message": "Could not find role: 0f3a2d418ed747fa8be46e92757be9ddff",
    "code": 404,
    "title": "Not Found"
  }
}
```

Status Codes

Status Code	Description
204	The request is successful. The agency has the specified permissions on the project.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.

Status Code	Description
500	Internal server error.

4.10.8 Querying the List of Permissions of an Agency on a Project

Function

This API is used to query the list of permissions of an agency on a project.

URI

- URI format
GET /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	ID of a project under the current domain.
agency_id	Yes	String	ID of an agency.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://v3.0/OS-AGENCY/projects/0945241c5ebc4660bac540d48f2a2c14/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
roles	Yes	Array	List of roles.

- Description for the role format

Parameter	Mandatory	Type	Description
id	Yes	String	ID of a role.
name	Yes	String	Name of a role.
domain_id	Yes	String	ID of the domain to which a role belongs.
type	Yes	String	Display mode of a role. <ul style="list-style-type: none"> • AX: A role is displayed at the domain layer. • XA: A role is displayed at the project layer. • AA: A role is displayed at both the domain and project layers. • XX: A role is not displayed at the domain or project layer.
display_name	Yes	String	Displayed name of a role.
catalog	Yes	String	Directory where a role locates.
policy	Yes	Dict	Policy of a role.
description	Yes	String	Description of a role.

- Example response (successful request)

```
{
  "roles": [
    {
      "catalog": "BASE",
      "display_name": "Tenant Guest",
      "name": "readonly",
      "policy": {
        "Version": "1.0",
        "Statement": [
          {
            "Action": [
              "::Get",
              "::List"
            ],
            "Effect": "Allow"
          },
          {
            "Action": [
              "identity:*"
            ],
            "Effect": "Deny"
          }
        ]
      }
    },
    {
      "domain_id": null,
      "type": "AA",
      "id": "b32d99a7778d4fd9aa5bc616c3dc4e5f",
      "description": "Tenant Guest"
    }
  ]
}
```

```
}
]
```

- Example response (request failed)

```
{
  "error": {
    "message": "You are not authorized to perform the requested action: identity:list_domain_grants",
    "code": 403,
    "title": "Forbidden"
  }
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.10.9 Deleting Permissions of an Agency on a Project

Function

This API is used to delete permissions of an agency on a project.

URI

- URI format
DELETE /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id}
- URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	ID of a project under the current domain.
agency_id	Yes	String	ID of an agency.
role_id	Yes	String	ID of a role.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json; charset=utf8' -X DELETE https://v3.0/OS-AGENCY/projects/0945241c5ebc4660bac540d48f2a2c14/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff
```

Response Parameters

- Example response (request failed)

```
{
  "error": {
    "message": "Could not find role: 0f3a2d418ed747fa8be46e92757be9dfff",
    "code": 404,
    "title": "Not Found"
  }
}
```

Status Codes

Status Code	Description
204	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.10.10 Granting Permissions to an Agency on a Domain

Function

This API is used to grant permissions to an agency on a domain.

URI

- URI format
PUT /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}
- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the current domain.
agency_id	Yes	String	ID of an agency.
role_id	Yes	String	ID of a role.

 **NOTE**

The role name corresponding to **role_id** in a request body is controlled by a blacklist. The role name cannot be **secu_admin** or **te_agency**.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json; charset=utf8' -X PUT https://v3.0/OS-AGENCY/domains/b32d99a7778d4fd9aa5bc616c3dc4e5f/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff
```

Response Parameters

- No response: indicates that the response is successful.
- Example response (request failed)

```
{
  "error": {
    "message": "Could not find role: 0f3a2d418ed747fa8be46e92757be9dfff",
    "code": 404,
    "title": "Not Found"
  }
}
```

Status Codes

Status Code	Description
204	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.

Status Code	Description
500	Internal server error.

4.10.11 Checking Whether an Agency Has the Specified Permissions on a Domain

Function

This API is used to check whether an agency has the specified permissions on a domain.

URI

- URI format
HEAD /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}
- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the current domain.
agency_id	Yes	String	ID of an agency.
role_id	Yes	String	ID of a role.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json; charset=utf8' -X HEAD https://v3.0/OS-AGENCY/domains/b32d99a7778d4fd9aa5bc616c3dc4e5f/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff
```

Response Parameters

- Example response (request failed)

```
{
  "error": {
    "message": "Could not find role: 0f3a2d418ed747fa8be46e92757be9dfff",
```

```
"code" : 404,
  "title" : "Not Found"
}
```

Status Codes

Status Code	Description
204	The request is successful. The agency has the specified permissions on the domain.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.10.12 Querying the List of Permissions of an Agency on a Domain

Function

This API is used to query the list of permissions of an agency on a domain.

URI

- URI format
GET /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles
- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the current domain.
agency_id	Yes	String	ID of an agency.

Request Parameters

- Parameters in the request header

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

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://v3.0/OS-AGENCY/domains/b32d99a7778d4fd9aa5bc616c3dc4e5f/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
roles	Yes	Array	List of roles.

- Description for the role format

Parameter	Mandatory	Type	Description
catalog	No	String	Directory where a role locates.
display_name	No	String	Displayed name of a role.
name	Yes	String	Name of a role.
policy	No	Dict	Policy of a role.
domain_id	Yes	String	ID of the domain to which a role belongs.
type	Yes	String	Display mode of a role. <ul style="list-style-type: none"> • AX: A role is displayed at the domain layer. • XA: A role is displayed at the project layer. • AA: A role is displayed at both the domain and project layers. • XX: A role is not displayed at the domain or project layer.
id	Yes	String	ID of a role.
description	No	String	Description of a role.

- Example response (successful request)

```
{
  "roles": [
    {
      "catalog": "BASE",
      "display_name": "Tenant Guest",
      "name": "readonly",
      "policy": {
        "Version": "1.0",
        "Statement": [
          {
            "Action": [
              "::Get",
              "::List"
            ],
            "Effect": "Allow"
          },
          {
            "Action": [
              "identity:*"
            ],
            "Effect": "Deny"
          }
        ]
      },
      "domain_id": null,
      "type": "AA",
      "id": "b32d99a7778d4fd9aa5bc616c3dc4e5f",
      "description": "Tenant Guest"
    }
  ]
}
```

- Example response (request failed)

```
{
  "error": {
    "message": "You are not authorized to perform the requested action: identity:list_domain_grants",
    "code": 403,
    "title": "Forbidden"
  }
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.10.13 Deleting Permissions of an Agency on a Domain

Function

This API is used to delete permissions of an agency on a domain.

URI

- URI format
DELETE /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}

- URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	ID of the current domain.
agency_id	Yes	String	ID of an agency.
role_id	Yes	String	ID of a role.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X DELETE https://v3.0/OS-AGENCY/domains/b32d99a7778d4fd9aa5bc616c3dc4e5f/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff
```

Response Parameters

- Example response (request failed)

```
{
  "error": {
    "message": "Could not find role: 0f3a2d418ed747fa8be46e92757be9dfff",
    "code": 404,
    "title": "Not Found"
  }
}
```

Status Codes

Status Code	Description
204	The request is successful.
401	Authentication failed.
403	Access denied.

Status Code	Description
404	The requested resource cannot be found.
500	Internal server error.

4.10.14 Querying All Permissions of an Agency

Function

This API is provided for the administrator to query all permissions that have been assigned to an agency.

URI

GET /v3.0/OS-INHERIT/domains/{domain_id}/agencies/{agency_id}/roles/
inherited_to_projects

Table 4-164 URI parameters

Parameter	Mandatory	Type	Description
agency_id	Yes	String	Agency ID.
domain_id	Yes	String	ID of the delegating account.

Request Parameters

Table 4-165 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

Table 4-166 Parameters in the response body

Parameter	Type	Description
roles	Array of objects	Permission information.
links	object	Resource link information.

Table 4-167 roles

Parameter	Type	Description
id	String	Permission ID.
links	object	Permission resource link.
name	String	Permission name.

Table 4-168 links

Parameter	Type	Description
self	String	Resource link.

Example Request

```
GET https://sample.domain.com/v3.0/OS-INHERIT/domains/{domain_id}/agencies/{agency_id}/roles/
inherited_to_projects
```

Example Response

Status code: 200

The request is successful.

```
{
  "roles": [
    {
      "name": "system_all_154",
      "links": {
        "self": "https://sample.domain.com/v3/roles/04570dfe267c45a3940e1ae9de868..."
      },
      "id": "04570dfe267c45a3940e1ae9de868..."
    },
    {
      "name": "test1_admin",
      "links": {
        "self": "https://sample.domain.com/v3/roles/1bf20f1adba94747a6e02e1be3810..."
      },
      "id": "1bf20f1adba94747a6e02e1be3810..."
    }
  ],
}
```

```
"links" : {
  "self" : "https://sample.domain.com/v3.0/OSHERIT/domains/05b09b4723001dc90f27c0008f8b1.../
agencies/08c6652e86801d234f01c00078308.../roles/inherited_to_projects"
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.10.15 Granting Specified Permissions to an Agency for All Projects

Function

This API is provided for the administrator to grant specified permissions to an agency for all projects.

URI

PUT /v3.0/OS-INHERIT/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}/inherited_to_projects

Table 4-169 URI parameters

Parameter	Man dator y	Type	Description
agency_id	Yes	String	Agency ID.
domain_id	Yes	String	Domain ID of the delegating party.
role_id	Yes	String	Permission ID.

Request Parameters

Table 4-170 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

None

Example Request

```
PUT https://sample.domain.com/v3.0/OS-INHERIT/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}/inherited_to_projects
```

Example Response

None

Status Codes

Status Code	Description
204	The authorization is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.10.16 Checking Whether an Agency Has Specified Permissions

Function

This API is provided for the administrator to check whether an agency has specified permissions.

URI

```
HEAD /v3.0/OS-INHERIT/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}/inherited_to_projects
```

Table 4-171 URI parameters

Parameter	Mandatory	Type	Description
agency_id	Yes	String	Agency ID.
domain_id	Yes	String	Domain ID of the delegating party.
role_id	Yes	String	Permission ID.

Request Parameters

Table 4-172 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

None

Example Request

```
HEAD https://sample.domain.com/v3.0/OS-INHERIT/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}/inherited_to_projects
```

Example Response

None

Status Codes

Status Code	Description
204	The request is successful. (The agency has the specified permissions.)
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.10.17 Removing Specified Permissions of an Agency in All Projects

Function

This API is provided for the administrator to remove the specified permissions of an agency in all projects.

URI

DELETE /v3.0/OS-INHERIT/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}/inherited_to_projects

Table 4-173 URI parameters

Parameter	Mandatory	Type	Description
agency_id	Yes	String	Agency ID.
domain_id	Yes	String	Domain ID of the delegating party.
role_id	Yes	String	Permission ID.

Request Parameters

Table 4-174 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

None

Example Request

```
DELETE https://sample.domain.com/v3.0/OS-INHERIT/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}/inherited_to_projects
```

Example Response

None

Status Codes

Status Code	Description
204	Permissions are removed successfully.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.11 Security Settings

4.11.1 Querying the Operation Protection Policy

Function

This API is used to query the operation protection policy.

URI

GET /v3.0/OS-SECURITYPOLICY/domains/{domain_id}/protect-policy

Table 4-175 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID.

Request Parameters

Table 4-176 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

Table 4-177 Parameters in the response body

Parameter	Type	Description
<code>protect_policy</code>	object	Operation protection policy.

Table 4-178 `protect_policy`

Parameter	Type	Description
<code>operation_protection</code>	Boolean	Indicates whether operation protection has been enabled. The value can be true or false .

Example Request

GET `https://sample.domain.com/v3.0/OS-SECURITYPOLICY/domains/{domain_id}/protect-policy`

Example Response

Status code: 200

The request is successful.

```
{
  "protect_policy" : {
    "operation_protection" : false
  }
}
```

Status code: 403

Access denied.

- Example 1

```
{
  "error_msg" : "You are not authorized to perform the requested action.",
  "error_code" : "IAM.0002"
}
```

- Example 2

```
{
  "error_msg" : "Policy doesn't allow %(actions)s to be performed.",
  "error_code" : "IAM.0003"
}
```

Status code: 404

The requested resource cannot be found.

```
{
  "error_msg" : "Could not find %(target)s: %(target_id)s.",
  "error_code" : "IAM.0004"
}
```

Status code: 500

Internal server error.

```
{
  "error_msg": "An unexpected error prevented the server from fulfilling your request.",
  "error_code": "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.11.2 Modifying the Operation Protection Policy

Function

This API is provided for the administrator to modify the operation protection policy.

URI

PUT /v3.0/OS-SECURITYPOLICY/domains/{domain_id}/protect-policy

Table 4-179 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID.

Request Parameters

Table 4-180 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Table 4-181 Parameters in the request body

Parameter	Mandatory	Type	Description
protect_policy	Yes	object	Operation protection policy.

Table 4-182 protect_policy

Parameter	Mandatory	Type	Description
operation_protection	Yes	Boolean	Indicates whether operation protection has been enabled. The value can be true or false .

Response Parameters

Table 4-183 Parameters in the response body

Parameter	Type	Description
protect_policy	object	Operation protection policy.

Table 4-184 protect_policy

Parameter	Type	Description
operation_protection	Boolean	Indicates whether operation protection has been enabled. The value can be true or false .

Example Request

```
PUT https://sample.domain.com/v3.0/OS-SECURITYPOLICY/domains/{domain_id}/protect-policy
{
  "protect_policy" : {
    "operation_protection" : true
  }
}
```

Example Response

Status code: 200

The request is successful.

```
{
  "protect_policy" : {
    "operation_protection" : false
  }
}
```

Status code: 400

The request body is abnormal.

- Example 1

```
{
  "error_msg" : "%(key)s' is a required property.",
  "error_code" : "IAM.0072"
}
```

- Example 2

```
{
  "error_msg" : "Invalid input for field '%(key)s'. The value is '%(value)s'.",
  "error_code" : "IAM.0073"
}
```

Status code: 403

Access denied.

- Example 1

```
{
  "error_msg" : "Policy doesn't allow %(actions)s to be performed.",
  "error_code" : "IAM.0003"
}
```

- Example 2

```
{
  "error_msg" : "You are not authorized to perform the requested action.",
  "error_code" : "IAM.0002"
}
```

Status code: 500

The system is abnormal.

```
{
  "error_msg" : "An unexpected error prevented the server from fulfilling your request.",
  "error_code" : "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The request body is abnormal.
401	Authentication failed.

Status Code	Description
403	Access denied.
500	The system is abnormal.

4.11.3 Querying the Password Policy

Function

This API is used to query the password policy.

URI

GET /v3.0/OS-SECURITYPOLICY/domains/{domain_id}/password-policy

Table 4-185 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID.

Request Parameters

Table 4-186 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

Table 4-187 Parameters in the response body

Parameter	Type	Description
password_policy	object	Password policy.

Table 4-188 password_policy

Parameter	Type	Description
maximum_consecutive_identical_chars	Integer	Maximum number of times that a character is allowed to consecutively present in a password.
maximum_password_length	Integer	Maximum number of characters that a password can contain.
minimum_password_age	Integer	Minimum period (minutes) after which users are allowed to make a password change.
minimum_password_length	Integer	Minimum number of characters that a password must contain.
number_of_recent_passwords_disallowed	Integer	Number of previously used passwords that are not allowed.
password_not_username_or_invert	Boolean	Indicates whether the password can be the username or the username spelled backwards.
password_requirements	String	Characters that a password must contain.
password_validity_period	Integer	Password validity period (days).

Example Request

GET https://sample.domain.com/v3.0/OS-SECURITYPOLICY/domains/{domain_id}/password-policy

Example Response

Status code: 200

The request is successful.

```
{
  "password_policy": {
    "password_requirements": "A password must contain at least two of the following: uppercase letters, lowercase letters, digits, and special characters.",
    "minimum_password_age": 20,
    "minimum_password_length": 8,
    "maximum_password_length": 32,
    "number_of_recent_passwords_disallowed": 2,
    "password_validity_period": 60,
    "maximum_consecutive_identical_chars": 3,
    "password_not_username_or_invert": true
  }
}
```

Status code: 403

Access denied.

- Example 1

```
{
  "error_msg": "You are not authorized to perform the requested action.",
  "error_code": "IAM.0002"
}
```

- Example 2

```
{
  "error_msg": "Policy doesn't allow %(actions)s to be performed.",
  "error_code": "IAM.0003"
}
```

Status code: 404

The requested resource cannot be found.

```
{
  "error_msg": "Could not find %(target)s: %(target_id)s.",
  "error_code": "IAM.0004"
}
```

Status code: 500

Internal server error.

```
{
  "error_msg": "An unexpected error prevented the server from fulfilling your request.",
  "error_code": "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.11.4 Modifying the Password Policy

Function

This API is provided for the administrator to modify the password policy.

URI

PUT /v3.0/OS-SECURITYPOLICY/domains/{domain_id}/password-policy

Table 4-189 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID.

Request Parameters

Table 4-190 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Table 4-191 Parameters in the request body

Parameter	Mandatory	Type	Description
password_policy	Yes	object	Password policy.

Table 4-192 password_policy

Parameter	Mandatory	Type	Description
maximum_consecutive_identical_chars	No	Integer	Maximum number of times that a character is allowed to consecutively present in a password. Value range: 0-32.
minimum_password_age	No	Integer	Minimum period (minutes) after which users are allowed to make a password change. Value range: 0-1440.
minimum_password_length	No	Integer	Minimum number of characters that a password must contain. Value range: 6-32.

Parameter	Mandatory	Type	Description
number_of_recent_passwords_disallowed	No	Integer	Number of previously used passwords that are not allowed. Value range: 0–10.
password_not_username_or_invert	No	Boolean	Indicates whether the password can be the username or the username spelled backwards.
password_validity_period	No	Integer	Password validity period (days). Value range: 0–180. Value 0 indicates that this requirement does not apply.

Response Parameters

Table 4-193 Parameters in the response body

Parameter	Type	Description
password_policy	object	Password policy.

Table 4-194 password_policy

Parameter	Type	Description
maximum_consecutive_identical_chars	Integer	Maximum number of times that a character is allowed to consecutively present in a password.
maximum_password_length	Integer	Maximum number of characters that a password can contain.
minimum_password_age	Integer	Minimum period (minutes) after which users are allowed to make a password change.
minimum_password_length	Integer	Minimum number of characters that a password must contain.
number_of_recent_passwords_disallowed	Integer	Number of previously used passwords that are not allowed.
password_not_username_or_invert	Boolean	Indicates whether the password can be the username or the username spelled backwards.

Parameter	Type	Description
password_requirements	String	Characters that a password must contain.
password_validity_period	Integer	Password validity period (days).

Example Request

PUT https://sample.domain.com/v3.0/OS-SECURITYPOLICY/domains/{domain_id}/password_policy

```
{
  "password_policy": {
    "minimum_password_length": 6,
    "number_of_recent_passwords_disallowed": 2,
    "minimum_password_age": 20,
    "password_validity_period": 60,
    "maximum_consecutive_identical_chars": 3,
    "password_not_username_or_invert": false
  }
}
```

Example Response

Status code: 200

The request is successful.

```
{
  "password_policy": {
    "password_requirements": "A password must contain at least two of the following: uppercase letters, lowercase letters, digits, and special characters.",
    "minimum_password_age": 20,
    "minimum_password_length": 8,
    "maximum_password_length": 32,
    "number_of_recent_passwords_disallowed": 2,
    "password_validity_period": 60,
    "maximum_consecutive_identical_chars": 3,
    "password_not_username_or_invert": true
  }
}
```

Status code: 400

The request body is abnormal.

- Example 1

```
{
  "error_msg": "%(key)s' is a required property.",
  "error_code": "IAM.0072"
}
```

- Example 2

```
{
  "error_msg": "Invalid input for field '%(key)s'. The value is '%(value)s'.",
  "error_code": "IAM.0073"
}
```

Status code: 403

Access denied.

- Example 1

```
{
  "error_msg": "You are not authorized to perform the requested action.",
  "error_code": "IAM.0002"
}
```

- Example 2

```
{
  "error_msg": "Policy doesn't allow %(actions)s to be performed.",
  "error_code": "IAM.0003"
}
```

Status code: 500

The system is abnormal.

```
{
  "error_msg": "An unexpected error prevented the server from fulfilling your request.",
  "error_code": "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The request body is abnormal.
401	Authentication failed.
403	Access denied.
500	The system is abnormal.

4.11.5 Querying the Login Authentication Policy

Function

This API is used to query the login authentication policy.

URI

GET /v3.0/OS-SECURITYPOLICY/domains/{domain_id}/login-policy

Table 4-195 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID.

Request Parameters

Table 4-196 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

Table 4-197 Parameters in the response body

Parameter	Type	Description
login_policy	object	Login authentication policy.

Table 4-198 login_policy

Parameter	Type	Description
account_validity_period	Integer	Validity period (days) to disable users if they have not logged in within the period. Value range: 0–240. Validity period (days) to disable users if they have not logged in within the period. If this parameter is set to 0 , no users will be disabled.
custom_info_for_login	String	Custom information that will be displayed upon successful login.
lockout_duration	Integer	Duration (minutes) to lock users out.
login_failed_times	Integer	Number of unsuccessful login attempts to lock users out.
period_with_login_failures	Integer	Period (minutes) to count the number of unsuccessful login attempts.
session_timeout	Integer	Session timeout (minutes) that will apply if you or users created using your account do not perform any operations within a specific period.
show_recent_login_info	Boolean	Indicates whether to display last login information upon successful login.

Example Request

```
GET https://sample.domain.com/v3.0/OS-SECURITYPOLICY/domains/{domain_id}/login-policy
```

Example Response

Status code: 200

The request is successful.

```
{
  "login_policy" : {
    "custom_info_for_login" : "",
    "period_with_login_failures" : 15,
    "lockout_duration" : 15,
    "account_validity_period" : 99,
    "login_failed_times" : 3,
    "session_timeout" : 16,
    "show_recent_login_info" : true
  }
}
```

Status code: 403

Access denied.

- Example 1

```
{
  "error_msg" : "You are not authorized to perform the requested action.",
  "error_code" : "IAM.0002"
}
```

- Example 2

```
{
  "error_msg" : "Policy doesn't allow %(actions)s to be performed.",
  "error_code" : "IAM.0003"
}
```

Status code: 404

The requested resource cannot be found.

```
{
  "error_msg" : "Could not find %(target)s: %(target_id)s.",
  "error_code" : "IAM.0004"
}
```

Status code: 500

Internal server error.

```
{
  "error_msg" : "An unexpected error prevented the server from fulfilling your request.",
  "error_code" : "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.

Status Code	Description
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.11.6 Modifying the Login Authentication Policy

Function

This API is provided for the administrator to modify the login authentication policy.

URI

PUT /v3.0/OS-SECURITYPOLICY/domains/{domain_id}/login-policy

Table 4-199 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID.

Request Parameters

Table 4-200 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Table 4-201 Parameters in the request body

Parameter	Mandatory	Type	Description
login_policy	Yes	object	Login authentication policy.

Table 4-202 login_policy

Parameter	Mandatory	Type	Description
account_validity_period	No	Integer	Validity period (days) to disable users if they have not logged in within the period. Value range: 0–240. If this parameter is set to 0 , no users will be disabled.
custom_info_for_login	No	String	Custom information that will be displayed upon successful login.
lockout_duration	No	Integer	Duration (minutes) to lock users out. Value range: 15–30.
login_failed_times	No	Integer	Number of unsuccessful login attempts to lock users out. Value range: 3–10.
period_with_login_failures	No	Integer	Period (minutes) to count the number of unsuccessful login attempts. Value range: 15–60.
session_timeout	No	Integer	Session timeout (minutes) that will apply if you or users created using your account do not perform any operations within a specific period. Value range: 15–1440.
show_recent_login_info	No	Boolean	Indicates whether to display last login information upon successful login. The value can be true or false .

Response Parameters

Table 4-203 Parameters in the response body

Parameter	Type	Description
login_policy	object	Login authentication policy.

Table 4-204 login_policy

Parameter	Type	Description
account_validity_period	Integer	Validity period (days) to disable users if they have not logged in within the period.

Parameter	Type	Description
custom_info_for_login	String	Custom information that will be displayed upon successful login.
lockout_duration	Integer	Duration (minutes) to lock users out.
login_failed_times	Integer	Number of unsuccessful login attempts to lock users out.
period_with_login_failures	Integer	Period (minutes) to count the number of unsuccessful login attempts.
session_timeout	Integer	Session timeout (minutes) that will apply if you or users created using your account do not perform any operations within a specific period.
show_recent_login_info	Boolean	Indicates whether to display last login information upon successful login.

Example Request

PUT https://sample.domain.com/v3.0/OS-SECURITYPOLICY/domains/{domain_id}/login-policy

```
{
  "login_policy": {
    "custom_info_for_login": "",
    "period_with_login_failures": 15,
    "lockout_duration": 15,
    "account_validity_period": 99,
    "login_failed_times": 3,
    "session_timeout": 16,
    "show_recent_login_info": true
  }
}
```

Example Response

Status code: 200

The request is successful.

```
{
  "login_policy": {
    "custom_info_for_login": "",
    "period_with_login_failures": 15,
    "lockout_duration": 15,
    "account_validity_period": 99,
    "login_failed_times": 3,
    "session_timeout": 16,
    "show_recent_login_info": true
  }
}
```

Status code: 400

The request body is abnormal.

- Example 1

```
{
  "error_msg": "'%(key)s' is a required property.",
  "error_code": "IAM.0072"
}
```

- Example 2

```
{
  "error_msg": "Invalid input for field '%(key)s'. The value is '%(value)s'.",
  "error_code": "IAM.0073"
}
```

Status code: 403

Access denied.

```
{
  "error_msg": "You are not authorized to perform the requested action.",
  "error_code": "IAM.0002"
}
```

Status code: 500

The system is abnormal.

```
{
  "error_msg": "An unexpected error prevented the server from fulfilling your request.",
  "error_code": "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The request body is abnormal.
401	Authentication failed.
403	Access denied.
500	The system is abnormal.

4.11.7 Querying the ACL for Console Access

Function

This API is used to query the ACL for console access.

URI

GET /v3.0/OS-SECURITYPOLICY/domains/{domain_id}/console-acl-policy

Table 4-205 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID.

Request Parameters

Table 4-206 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

Table 4-207 Parameters in the response body

Parameter	Type	Description
console_acl_policy	object	ACL for console access.

Table 4-208 console_acl_policy

Parameter	Type	Description
allow_addresses_netmasks	Array of objects	IPv4 CIDR blocks from which console access is allowed.
allow_ip_ranges	Array of objects	IP address ranges from which console access is allowed.

Table 4-209 allow_address_netmasks

Parameter	Type	Description
address_netmask	String	IPv4 CIDR block, for example, 192.168.0.1/24 .
description	String	Description about the IPv4 CIDR block.

Table 4-210 allow_ip_ranges

Parameter	Type	Description
description	String	Description about an IP address range.
ip_range	String	IP address range, for example, 0.0.0.0-255.255.255.255 .

Example Request

GET https://sample.domain.com/v3.0/OS-SECURITYPOLICY/domains/{domain_id}/console-acl-policy

Example Response

Status code: 200

The request is successful.

```
{
  "console_acl_policy": {
    "allow_ip_ranges": [ {
      "ip_range": "0.0.0.0-255.255.255.255",
      "description": ""
    }, {
      "ip_range": "0.0.0.0-255.255.255.255",
      "description": ""
    } ],
    "allow_address_netmasks": [ {
      "address_netmask": "192.168.0.1/24",
      "description": ""
    }, {
      "address_netmask": "192.168.0.1/24",
      "description": ""
    } ]
  }
}
```

Status code: 403

Access denied.

- Example 1

```
{
  "error_msg": "You are not authorized to perform the requested action.",
  "error_code": "IAM.0002"
}
```

- Example 2

```
{
  "error_msg": "Policy doesn't allow %(actions)s to be performed.",
  "error_code": "IAM.0003"
}
```

Status code: 404

The requested resource cannot be found.

```
{
  "error_msg": "Could not find %(target)s: %(target_id)s.",
  "error_code": "IAM.0004"
}
```

Status code: 500

Internal server error.

```
{
  "error_msg" : "An unexpected error prevented the server from fulfilling your request.",
  "error_code" : "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.11.8 Modifying the ACL for Console Access

Function

This API is provided for the administrator to modify the ACL for console access.

URI

PUT /v3.0/OS-SECURITYPOLICY/domains/{domain_id}/console-acl-policy

Table 4-211 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID.

Request Parameters

Table 4-212 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Table 4-213 Parameters in the request body

Parameter	Mandatory	Type	Description
console_acl_policy	Yes	object	ACL for console access.

Table 4-214 console_acl_policy

Parameter	Mandatory	Type	Description
allow_addresses_netmasks	Yes	Array of objects	IPv4 CIDR blocks from which console access is allowed.
allow_ip_ranges	Yes	Array of objects	IP address ranges from which console access is allowed.

Table 4-215 allow_address_netmasks

Parameter	Mandatory	Type	Description
address_netmask	Yes	String	IPv4 CIDR block, for example, 192.168.0.1/24 .
description	No	String	Description about the IPv4 CIDR block.

Table 4-216 allow_ip_ranges

Parameter	Mandatory	Type	Description
description	No	String	Description about an IP address range.
ip_range	Yes	String	IP address range, for example, 0.0.0.0-255.255.255.255 .

Response Parameters

Table 4-217 Parameters in the response body

Parameter	Type	Description
console_acl_policy	object	ACL for console access.

Table 4-218 console_acl_policy

Parameter	Type	Description
allow_addresses_netmasks	Array of objects	IPv4 CIDR blocks from which console access is allowed.
allow_ip_ranges	Array of objects	IP address ranges from which console access is allowed.

Table 4-219 allow_address_netmasks

Parameter	Type	Description
address_netmask	String	IPv4 CIDR block, for example, 192.168.0.1/24 .
description	String	Description about the IPv4 CIDR block.

Table 4-220 allow_ip_ranges

Parameter	Type	Description
description	String	Description about an IP address range.
ip_range	String	IP address range, for example, 0.0.0.0-255.255.255.255 .

Example Request

```
PUT https://sample.domain.com/v3.0/OS-SECURITYPOLICY/domains/{domain_id}/console-acl-policy
```

```
{
  "console_acl_policy": {
    "allow_ip_ranges": [ {
      "ip_range": "0.0.0.0-255.255.255.255",
      "description": "1"
    }, {
      "ip_range": "0.0.0.0-255.255.255.253",
      "description": "12"
    } ],
    "allow_address_netmasks": [ {
```

```
"address_netmask" : "192.168.0.1/24",  
"description" : "3"  
}, {  
"address_netmask" : "192.168.0.2/23",  
"description" : "4"  
}  
}]  
}
```

Example Response

Status code: 200

The request is successful.

```
{  
"console_acl_policy" : {  
"allow_ip_ranges" : [ {  
"ip_range" : "0.0.0.0-255.255.255.255",  
"description" : ""  
}, {  
"ip_range" : "0.0.0.0-255.255.255.255",  
"description" : ""  
}],  
"allow_address_netmasks" : [ {  
"address_netmask" : "192.168.0.1/24",  
"description" : ""  
}, {  
"address_netmask" : "192.168.0.1/24",  
"description" : ""  
}]  
}  
}
```

Status code: 400

The request body is abnormal.

- Example 1

```
{  
"error_msg" : "%(key)s' is a required property.",  
"error_code" : "IAM.0072"  
}
```

- Example 2

```
{  
"error_msg" : "Invalid input for field '%(key)s'. The value is '%(value)s'.",  
"error_code" : "IAM.0073"  
}
```

Status code: 500

The system is abnormal.

```
{  
"error_msg" : "An unexpected error prevented the server from fulfilling your request.",  
"error_code" : "IAM.0006"  
}
```

Status Codes

Status Code	Description
200	The request is successful.

Status Code	Description
400	The request body is abnormal.
401	Authentication failed.
403	Access denied.
500	The system is abnormal.

4.11.9 Querying the ACL for API Access

Function

This API is used to query the ACL for API access.

URI

GET /v3.0/OS-SECURITYPOLICY/domains/{domain_id}/api-acl-policy

Table 4-221 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID.

Request Parameters

Table 4-222 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Response Parameters

Table 4-223 Parameters in the response body

Parameter	Type	Description
api_acl_policy	object	ACL for API access.

Table 4-224 [api_acl_policy](#)

Parameter	Type	Description
allow_addresses_netmasks	Array of objects	IPv4 CIDR blocks from which API access is allowed.
allow_ip_ranges	Array of objects	IP address ranges from which API access is allowed.

Table 4-225 [allow_address_netmasks](#)

Parameter	Type	Description
address_netmask	String	IPv4 CIDR block, for example, 192.168.0.1/24 .
description	String	Description about the IPv4 CIDR block.

Table 4-226 [allow_ip_ranges](#)

Parameter	Type	Description
description	String	Description about an IP address range.
ip_range	String	IP address range, for example, 0.0.0.0-255.255.255.255 .

Example Request

```
GET https://sample.domain.com/v3.0/OS-SECURITYPOLICY/domains/{domain_id}/api-acl-policy
```

Example Response

Status code: 200

The request is successful.

```
{
  "api_acl_policy" : {
```

```

"allow_ip_ranges" : [ {
  "ip_range" : "0.0.0.0-255.255.255.255",
  "description" : ""
}, {
  "ip_range" : "0.0.0.0-255.255.255.255",
  "description" : ""
} ],
"allow_address_netmasks" : [ {
  "address_netmask" : "192.168.0.1/24",
  "description" : ""
}, {
  "address_netmask" : "192.168.0.1/24",
  "description" : ""
} ]
}
}

```

Status code: 403

Access denied.

- Example 1

```

{
  "error_msg" : "You are not authorized to perform the requested action.",
  "error_code" : "IAM.0002"
}

```

- Example 2

```

{
  "error_msg" : "Policy doesn't allow %(actions)s to be performed.",
  "error_code" : "IAM.0003"
}

```

Status code: 404

The requested resource cannot be found.

```

{
  "error_msg" : "Could not find %(target)s: %(target_id)s.",
  "error_code" : "IAM.0004"
}

```

Status code: 500

Internal server error.

```

{
  "error_msg" : "An unexpected error prevented the server from fulfilling your request.",
  "error_code" : "IAM.0006"
}

```

Status Codes

Status Code	Description
200	The request is successful.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
500	Internal server error.

4.11.10 Modifying the ACL for API Access

Function

This API is provided for the administrator to modify the ACL for API access.

URI

PUT /v3.0/OS-SECURITYPOLICY/domains/{domain_id}/api-acl-policy

Table 4-227 URI parameters

Parameter	Mandatory	Type	Description
domain_id	Yes	String	Domain ID.

Request Parameters

Table 4-228 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Token with Security Administrator permissions.

Table 4-229 Parameters in the request body

Parameter	Mandatory	Type	Description
api_acl_policy	Yes	object	ACL for API access.

Table 4-230 api_acl_policy

Parameter	Mandatory	Type	Description
allow_addresses_netmasks	Yes	Array of objects	IPv4 CIDR blocks from which API access is allowed.
allow_ip_ranges	Yes	Array of objects	IP address ranges from which API access is allowed.

Table 4-231 allow_address_netmasks

Parameter	Mandatory	Type	Description
address_netmask	Yes	String	IPv4 CIDR block, for example, 192.168.0.1/24 .
description	No	String	Description about the IPv4 CIDR block.

Table 4-232 allow_ip_ranges

Parameter	Mandatory	Type	Description
description	No	String	Description about an IP address range.
ip_range	Yes	String	IP address range, for example, 0.0.0.0-255.255.255.255 .

Response Parameters

Table 4-233 Parameters in the response body

Parameter	Type	Description
api_acl_policy	object	ACL for API access.

Table 4-234 api_acl_policy

Parameter	Type	Description
allow_addresses_netmasks	objects	IPv4 CIDR blocks from which API access is allowed.
allow_ip_ranges	objects	IP address ranges from which API access is allowed.

Table 4-235 allow_address_netmasks

Parameter	Type	Description
address_netmask	String	IPv4 CIDR block, for example, 192.168.0.1/24 .
description	String	Description about the IPv4 CIDR block.

Table 4-236 allow_ip_ranges

Parameter	Type	Description
description	String	Description about an IP address range.
ip_range	String	IP address range, for example, 0.0.0.0-255.255.255.255 .

Example Request

```
PUT https://sample.domain.com/v3.0/OS-SECURITYPOLICY/domains/{domain_id}/api-acl-policy
{
  "api_acl_policy": {
    "allow_ip_ranges": [ {
      "ip_range": "0.0.0.0-255.255.255.255",
      "description": "1"
    }, {
      "ip_range": "0.0.0.0-255.255.255.253",
      "description": "12"
    } ],
    "allow_address_netmasks": [ {
      "address_netmask": "192.168.0.1/24",
      "description": "3"
    }, {
      "address_netmask": "192.168.0.2/23",
      "description": "4"
    } ]
  }
}
```

Example Response

Status code: 200

The request is successful.

```
{
  "api_acl_policy" : {
    "allow_ip_ranges" : [ {
      "ip_range" : "0.0.0.0-255.255.255.255",
      "description" : ""
    }, {
      "ip_range" : "0.0.0.0-255.255.255.255",
      "description" : ""
    } ],
    "allow_address_netmasks" : [ {
      "address_netmask" : "192.168.0.1/24",
      "description" : ""
    }, {
      "address_netmask" : "192.168.0.1/24",
      "description" : ""
    } ]
  }
}
```

Status code: 400

The request body is abnormal.

- Example 1

```
{
  "error_msg" : "%(key)s' is a required property.",
  "error_code" : "IAM.0072"
}
```

- Example 2

```
{
  "error_msg" : "Invalid input for field '%(key)s'. The value is '%(value)s'.",
  "error_code" : "IAM.0073"
}
```

Status code: 500

The system is abnormal.

```
{
  "error_msg" : "An unexpected error prevented the server from fulfilling your request.",
  "error_code" : "IAM.0006"
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The request body is abnormal.
401	Authentication failed.
403	Access denied.
500	The system is abnormal.

4.12 Federated Identity Authentication Management

4.12.1 Obtaining a Token in Federated Identity Authentication Mode

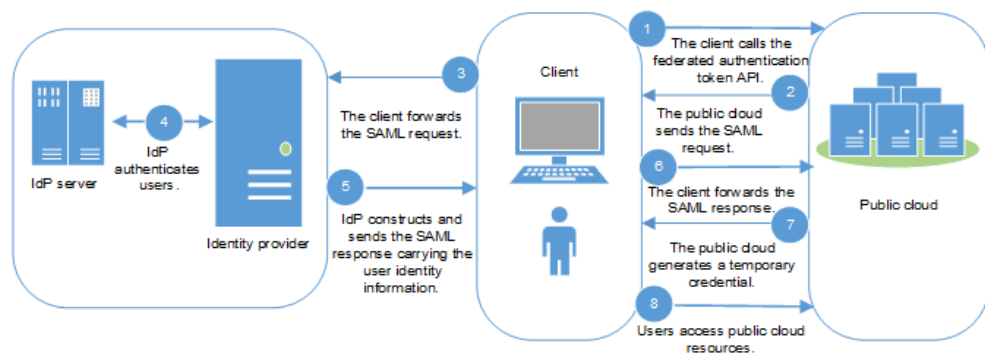
4.12.1.1 SP Initiated

OpenStack and Shibboleth are widely used open-source federated identity authentication solutions. They provide powerful SSO capabilities and connect users to various applications both inside and outside enterprises. This section describes how to use OpenStackClient and Shibboleth ECP Client to obtain the federated authentication token.

Flowchart

The following figure shows the SP-initiated federation authentication process.

Figure 4-1 Flowchart (SP-initiated)



Description

1. The client calls the API (federated token obtained in the SP-initiated mode) provided by the cloud system.
2. The cloud system searches for the metadata file based on the user and IdP information in the URL and sends the SAML request to the client.
3. The client encapsulates the SAML request and forwards the SAML request to the IdP.
4. A user enters a username and password on the IdP server for identity authentication.
5. After the user passes the authentication, IdP constructs an assertion carrying the user identity information and sends the SAML response. The response passes through the client.
6. The client encapsulates the SAML response and forwards the SAML response to the cloud platform.

7. The cloud platform verifies and authenticates the assertion, and generates a temporary access credential according to the identity conversion rule configured by users in the identity provider.
8. Users can access cloud resources according to their permissions.

OpenStackClient

You must have permissions of user **root** to install the unified command-line client. To perform the following operations, you only need to have the permissions of a common user.

NOTICE

The API calling operation must be performed in a secure network environment (in a VPN or a cloud server of a domain). Otherwise, this operation may be under the man-in-the-middle (MITM) attack.

- Step 1** Create an environment variable file under the installation directory of OpenStackClient. Modify the environment variable file in a text editor. Add parameters, such as the username, password, region, SAML protocol version, and the IP address and port number of IAM, to the file. [Table 4-237](#) describes the parameters.

For example:

```
export OS_IDENTITY_API_VERSION=3
export OS_AUTH_TYPE=v3samlpassword
export OS_IDENTITY_PROVIDER=idpid
export OS_PROTOCOL=saml
export OS_IDENTITY_PROVIDER_URL=https://idp.example.com/idp/profile/SAML2/SOAP/ECP
export OS_USERNAME=username
export OS_PASSWORD=userpassword
export OS_DOMAIN_NAME=example-domain-name
```

Table 4-237 Parameter description

Parameter	Description
OS_IDENTITY_API_VERSION	Indicates the authentication API version. The value is fixed at 3 .
OS_AUTH_TYPE	Indicates the authentication type. The value is fixed at v3samlpassword .

Parameter	Description
OS_AUTH_URL	Indicates the authentication URL. The value format is https://IAM IP address:Port number/API version . <ul style="list-style-type: none"> • <i>Port number</i> is fixed at 443. • <i>API version</i> is fixed at v3.
OS_IDENTITY_PROVIDER	Indicates the name of an identity provider created by a user in the cloud system. For example: Publiccloud-Shibboleth.
OS_DOMAIN_NAME	Indicates the domain name to be authenticated.
OS_PROTOCOL	Indicates the SAML protocol version. The value is fixed at saml .
OS_IDENTITY_PROVIDER_URL	Indicates the URL of the identity provider used to handle the authentication request initialized by the ECP.
OS_USERNAME	Indicates the name of a user who is authenticated in the identity provider.
OS_PASSWORD	Indicates the password of a user who is authenticated in the identity provider.

Step 2 Run the following command to set environment variables:

```
source keystonerc
```

Step 3 Run the following command to obtain a token:

```
openstack token issue
```

```
>>openstack token issue
command: token issue -> openstackclient.identity.v3.token.IssueToken (auth=True)
Using auth plugin: v3samlpassword
+-----+
| Field | Value
| expires | 2018-04-16T03:46:51+0000
| id      | MIIDbQYJKoZIhvcNAQcCoIIDXjXXX...
| user_id | 9B7Cjy5ME14f0fQKhb6HJVQdpXXX...
```

In the command output, **id** is the obtained federated authentication token.

----End

Shibboleth ECP Client

Step 1 Configure the **metadata-providers.xml** file in Shibboleth IdP v3 and save the **metadata.xml** file in the corresponding path.

```
<MetadataProvider id="LocalMetadata1" xsi:type="FilesystemMetadataProvider" metadataFile="C:\Program Files (x86)\Shibboleth\IdP\metadata\web_metadata.xml"/>
<MetadataProvider id="LocalMetadata2" xsi:type="FilesystemMetadataProvider" metadataFile="C:\Program Files (x86)\Shibboleth\IdP\metadata\api_metadata.xml"/>
```

 NOTE

- **MetadataProvider id** indicates the name of the downloaded metadata file of the SP system.
- **metadataFile** indicates the path for storing the metadata file of the SP system in the enterprise IdP.

Step 2 Configure the **attribute-filter.xml** file in Shibboleth IdP v3.

```
<afp:AttributeFilterPolicy id="example1">
  <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://
auth.example.com/" />
  <afp:AttributeRule attributeID="eduPersonPrincipalName">
    <afp:PermitValueRule xsi:type="basic:ANY" />
  </afp:AttributeRule>
  <afp:AttributeRule attributeID="uid">
    <afp:PermitValueRule xsi:type="basic:ANY" />
  </afp:AttributeRule>
  <afp:AttributeRule attributeID="mail">
    <afp:PermitValueRule xsi:type="basic:ANY" />
  </afp:AttributeRule>
</afp:AttributeFilterPolicy>

<afp:AttributeFilterPolicy id="example2">
  <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://iam.
{region_id}.example.com" />
  <afp:AttributeRule attributeID="eduPersonPrincipalName">
    <afp:PermitValueRule xsi:type="basic:ANY" />
  </afp:AttributeRule>
  <afp:AttributeRule attributeID="uid">
    <afp:PermitValueRule xsi:type="basic:ANY" />
  </afp:AttributeRule>
  <afp:AttributeRule attributeID="mail">
    <afp:PermitValueRule xsi:type="basic:ANY" />
  </afp:AttributeRule>
</afp:AttributeFilterPolicy>
```

 NOTE

AttributeFilterPolicy id indicates the name of the downloaded metadata file of the SP system.

value indicates the **EntityID** in the metadata file of the SP system.

Step 3 Configure the endpoint address of the enterprise IdP in the **ecp.py** script.

```
# mapping from user friendly names or tags to IdP ECP endpoints
IDP_ENDPOINTS = {
  "idp1": "https://idp.example.com/idp/profile/SAML2/SOAP/ECP"
}
```

Step 4 Run the **ecp.py** script to obtain the federated authentication token.

```
>>>python ecp.py
Usage: ecp.py [options] IdP_tag target_url login
>>>python ecp.py -d idp1 https://iam.{region_id}.example.com/v3/OS-FEDERATION/identity_providers/
idp_example/protocols/saml/auth {username}
X-Subject-Token: MIIDbQYJKoZIhvcNAQcColIDXXX...
```

X-Subject-Token is the obtained federated authentication token.

----End

4.12.1.2 IdP Initiated

This section uses the **Client4ShibbolethIdP** script as an example to describe how to obtain a federated authentication token in the IdP-initiated mode. The **Client4ShibbolethIdP** script simulates a user who logs in to the enterprise IdP

using a browser. Therefore, by comparing the form data submitted by the browser and the client implementation data, this section helps users develop the client scripts of their enterprise IdP.

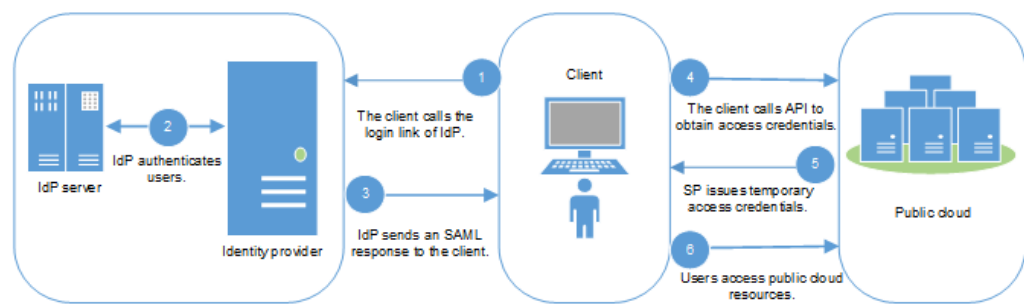
Prerequisites

- IdP-initiated federated identity authentication is supported by the enterprise IdP server.
- The **beautifulsoup4** package of the Python module has been installed on the client.

Flowchart

The following figure shows the IdP-initiated federation authentication process.

Figure 4-2 Flowchart (IdP-initiated)



Description

1. The client calls the login link provided by IdP based on the IdP-initiated mode and sets the cloud platform address in the login link, that is, **entityID** in the metadata file of the cloud platform.
2. The client obtains the login page of the IdP. Users submit identity information to IdP for authentication through the client.
3. After users pass the authentication, IdP constructs an assertion carrying the user identity information and sends the SAML response. The response passes through the client.
4. The client encapsulates the SAML response, forwards the SAML response, and calls the API (federated token obtained in the IdP-initiated mode) provided by the cloud platform.
5. The cloud platform verifies and authenticates the assertion, and generates a temporary access credential according to the identity conversion rule configured by users in the identity provider.
6. Users can access cloud resources according to their permissions.

Implementation on the Client

Step 1 Configure the login URL of enterprise IdP.

Table 4-238 Login URLs of common IdP products

IdP	SP Identification Parameter in URL	Login URL Example
ADFS	logintorp	https://adfs-server.contoso.com/adfs/ls/IdpInitiatedSignon.aspx?logintorp=https://iam.example.com
Shibboleth	providerId	https://idp.example.org/idp/profile/SAML2/Unsolicited/SSO?providerId=iam.example.com
SimpleSAMLphp	spentityid	https://idp.example.org/simplesaml/saml2/idp/SSOService.php?spentityid=iam.example.com

After the configuration, enter the login URL in the browser address box. The following page is displayed.

Figure 4-3 Login Page

Our Identity Provider
(replace this placeholder with your organizational logo / label)

Username

Password

> Forgot your password?
 > Need Help?

Don't Remember Login

Clear prior granting of permission for release of your information to this service.

Login

Client4ShibbolethIdP script implementation:

```
import sys
import requests
import getpass
import re
from bs4 import BeautifulSoup
from urlparse import urlparse

# SSL certificate verification: Whether or not strict certificate
# verification is done, False should only be used for dev/test
sslverification = True

# Get the federated credentials from the user
```

```

print "Username:",
username = raw_input()
password = getpass.getpass()
print "

session = requests.Session()

# The initial url that starts the authentication process.
idp_entry_url = 'https://idp.example.com/idp/profile/SAML2/Unsolicited/SSO?providerId=https://iam.example.com'

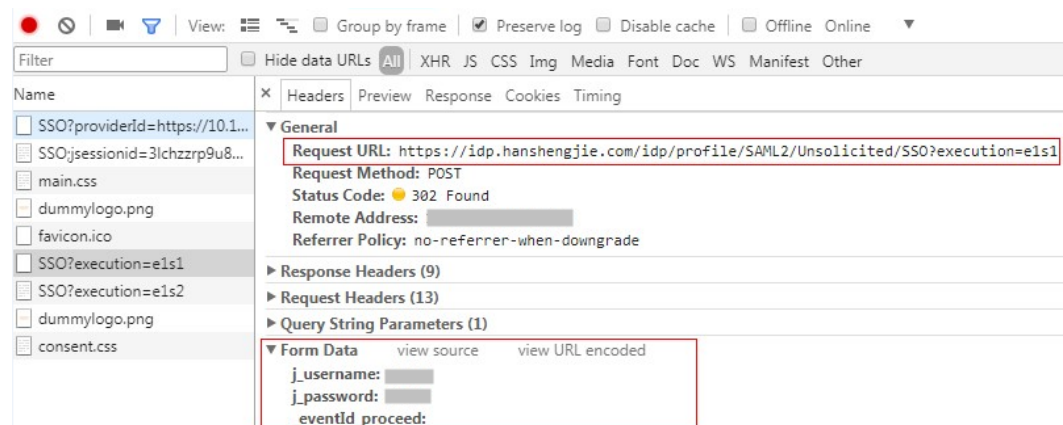
# Programmatically get the SAML assertion,open the initial IdP url# and follows all of the HTTP302
redirects, and gets the resulting# login page
formresponse = session.get(idp_entry_url, verify=sslverification)
# Capture the idp_authform_submit_url,which is the final url after# all the 302s
idp_authform_submit_url = formresponse.url

```

Step 2 The client submits authentication information. The client parses the login page using the beautifulsoup4 module, captures the user information input box and requested action, constructs the request parameters, and initiates identity authentication to the IdP.

Obtain all form data submitted for the login page from the browser.

Figure 4-4 Authentication information (1)



Client4ShibbolethIdP script implementation:

```

# Parse the response and extract all the necessary values in order to build a dictionary of all of the form
values the IdP expects
formsoup = BeautifulSoup(formresponse.text.decode('utf8'), "lxml")
payload = {}

for inputtag in formsoup.find_all(re.compile('(INPUT|input)')):
    name = inputtag.get('name', "")
    value = inputtag.get('value', "")
    if "username" in name.lower():
        payload[name] = username
    elif "password" in name.lower():
        payload[name] = password
    else:
        payload[name] = value

for inputtag in formsoup.find_all(re.compile('(FORM|form)')):
    action = inputtag.get('action')
    if action:
        parsedurl = urlparse(idp_entry_url)
        idp_authform_submit_url = parsedurl.scheme + "://" + parsedurl.netloc + action

```

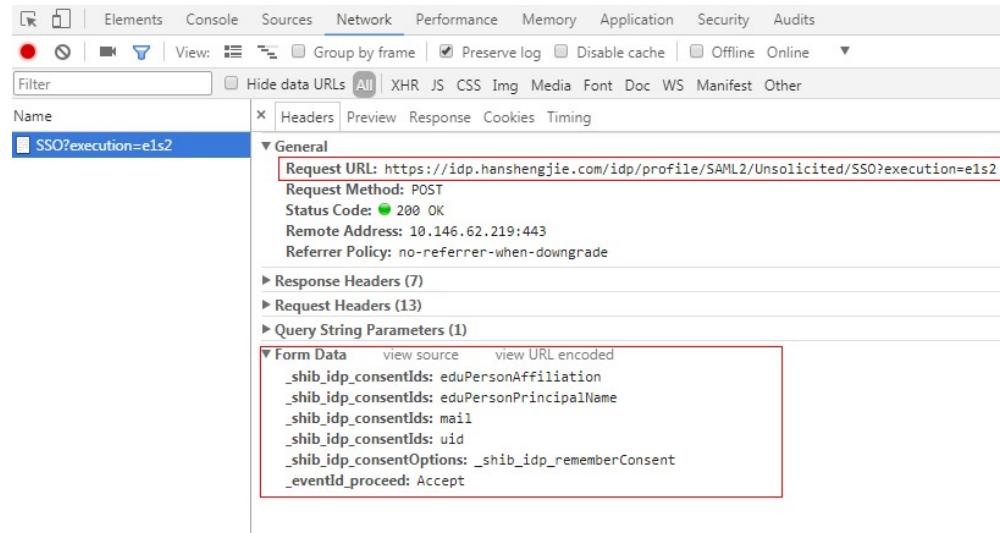
```
# please test on browser first, add other parameters in payload
payload["_eventId_proceed"] = ""

formresponse = session.post(
    idp_authform_submit_url, data=payload, verify=sslverification)
```

Step 3 The client parses the next page. (Some enterprise IdPs provide pages containing user attributes.)

Obtain all form data submitted for the login page from the browser.

Figure 4-5 Authentication information (2)



Client4ShibbolethIdP script implementation:

```
# In shebbleth IdP v3, browser will show attributes page for user,# so we need parse the page
formsoup = BeautifulSoup(formresponse.text.decode('utf8'), "xml")
payload = {}

# Add other form data required from browser to payload
_shib_idp_consents = []
for inputtag in formsoup.find_all(re.compile('input')):
    name = inputtag.get("name")
    value = inputtag.get("value")
    if name == "_shib_idp_consents":
        _shib_idp_consents.append(value)
payload["_shib_idp_consents"] = _shib_idp_consents
payload["_shib_idp_consentsOptions"] = "_shib_idp_rememberConsent"
payload["_eventId_proceed"] = "Accept"

# user can get the action url from the html file
nexturl = "https://idp.example.com/idp/profile/SAML2/Unsolicited/SSO?execution=e1s2"

for inputtag in formsoup.find_all(re.compile('(FORM|form)')):
    action = inputtag.get('action')
    if action:
        parsedurl = urlparse(idp_entry_url)
        nexturl = parsedurl.scheme + "://" + parsedurl.netloc + action

response = session.post(
    nexturl, data=payload, verify=sslverification)
```

Step 4 The client parses the response message sent from the IdP. The client submits user information to the enterprise IdP for authentication. After authenticating the user information, the IdP sends a response message to the client. The client parses the **SAMLResponse** parameter in the response message.

Client4ShibbolethIdP script implementation:

```
# Decode the response and extract the SAML assertion
soup = BeautifulSoup(response.text.decode('utf8'), "lxml")
SAMLResponse = ""

# Look for the SAMLResponse attribute of the input tag
for inputtag in soup.find_all('input'):
    if (inputtag.get('name') == 'SAMLResponse'):
        SAMLResponse = inputtag.get('value')

# Better error handling is required for production use.
if (SAMLResponse == ""):
    print 'Response did not contain a valid SAML assertion, please troubleshooting in Idp side.'
    sys.exit(0)
```

Step 5 Obtain an unscoped token. For details, see [4.12.6.2 Obtaining an Unscoped Token \(IdP Initiated\)](#).

Client4ShibbolethIdP script implementation:

```
# Set headers
headers = {}
headers["X-Idp-Id"] = "test_local_idp"

# IAM API url: get unscoped token on IDP initiated mode
sp_unscoped_token_url = "https://iam.example.com/v3.0/OS-FEDERATION/tokens"

# Set form data
payload = {}
payload["SAMLResponse"] = SAMLResponse
response = session.post(
    sp_unscoped_token_url, data=payload, headers=headers, verify=sslverification)

# Debug only
print(response.text)
print "Status Code: " + str(response.status_code)
if response.status_code != 201:
    sys.exit(1)

unscoped_token = response.headers.get("X-Subject-Token") if "X-Subject-Token" in response.headers.keys()
else None
if unscoped_token:
    print ">>>>>>X-Subject-Token: " + unscoped_token
```

Step 6 Obtain a scoped token. For details, see [4.12.6.3 Obtaining a Scoped Token](#).

Client4ShibbolethIdP script implementation:

```
payload = {
    "auth": {
        "identity": {
            "methods": ["token"],
            "token": {
                "id": unscoped_token
            }
        },
        "scope": {
            "project": {
                "name": "{region_id}_test1"
            }
        }
    }
}

sp_scoped_token_url = "https://iam.example.com/v3/auth/tokens"

response = session.post(
    sp_scoped_token_url, json=payload, verify=sslverification)
```

```
# Debug only
print "Status Code: " + str(response.status_code)
if response.status_code != 201:
    print response.text
    sys.exit(1)

scoped_token = response.text if response.status_code == 201 else None
if scoped_token:
    print ">>>>>>Scoped Token:" + scoped_token
```

Step 7 Obtain a temporary AK/SK.

Client4ShibbolethIdP script implementation:

```
# Set form data
payload = {
    "auth": {
        "identity": {
            "methods": ["token"],
            "token": {
                "duration_seconds": "900"
            }
        }
    }
}

# Set headers
headers = {}
headers["X-Auth-Token"] = unscoped_token

sp_STS_token_url = "https://iam.example.com/v3.0/OS-CREDENTIAL/securitytokens"

response = session.post(
    sp_STS_token_url, json=payload, headers=headers, verify=sslverification)

# Debug only
print "Status Code: " + str(response.status_code)
if response.status_code != 201:
    print response.text
    sys.exit(1)

sts_token = response.text if response.status_code == 201 else None
if sts_token:
    print ">>>>>>STS Token:" + sts_token
```

----End

4.12.2 Identity Provider

4.12.2.1 Querying the Identity Provider List

Function

This API is used to query the identity provider list.

URI

GET /v3/OS-FEDERATION/identity_providers

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.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/OS-FEDERATION/identity_providers
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
identity_providers	Yes	Array	List of identity providers.
links	Yes	Object	Identity provider resource link.

- Description for the identity_providers format

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an identity provider.
description	Yes	String	Identity provider description.
enabled	Yes	Boolean	Whether an identity provider is enabled. true indicates that the identity provider is enabled. false indicates that the identity provider is disabled. The default value is false .
remote_ids	Yes	Array	Federated user ID list of an identity provider.
links	Yes	Object	Identity provider resource link.

- Example response

```
{
  "identity_providers": [
    {
      "description": "Stores ACME identities",
      "enabled": true,
      "id": "ACME",

```

```

    "remote_ids": [],
    "links": {
      "protocols": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME/protocols",
      "self": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME"
    }
  },
  {
    "description": "Stores contractor identities",
    "enabled": false,
    "remote_ids": [],
    "id": "ACME-contractors",

    "links": {
      "protocols": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME-
contractors/protocols",
      "self": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME-contractors"
    }
  }
],
"links": {
  "next": null,
  "previous": null,
  "self": "https://sample.domain.com/v3/OS-FEDERATION/identity_providers"
}
}

```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.2.2 Querying an Identity Provider

Function

This API is used to query the information about an identity provider.

URI

- URI format
GET /v3/OS-FEDERATION/identity_providers/{id}
- URI parameters

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an identity provider.

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.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H 'X-Auth-Token:$token' -X GET https://sample.domain.com/v3/OS-FEDERATION/identity_providers/ACME
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an identity provider.
description	Yes	String	Identity provider description.
enabled	Yes	Boolean	Whether an identity provider is enabled. <ul style="list-style-type: none"> • true indicates that the identity provider is enabled. • false indicates that the identity provider is disabled. The default value is false .
remote_ids	Yes	Array	Federated user ID list of an identity provider.
links	Yes	Object	Identity provider resource link.

- Example response

```
{
  "identity_provider": {
```

```

    "description": "Stores ACME identities",
    "enabled": false,
    "id": "ACME",

    "remote_ids": [],
    "links": {
      "protocols": "https://sample.domain.com/v3/OS-FEDERATION/identity_providers/ACME/protocols",
      "self": "https://sample.domain.com/v3/OS-FEDERATION/identity_providers/ACME"
    }
  }
}

```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.2.3 Creating a SAML Identity Provider

Function

This API is used to create a SAML identity provider.

URI

- URI format
PUT /v3/OS-FEDERATION/identity_providers/{id}
- URI parameters

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an identity provider.

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
identity_provider	Yes	JSON object	Request body for registering an identity provider.
description	No	String	Identity provider description.
enabled	No	Boolean	Whether an identity provider is enabled. <ul style="list-style-type: none"> true indicates that the identity provider is enabled. false indicates that the identity provider is disabled. The default value is false .

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H 'X-Auth-Token:$token' -X PUT -d '{"identity_provider":{"description":"Stores ACME identities.", "enabled":true}}' https://sample.domain.com/v3/OS-FEDERATION/identity_providers/ACME
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an identity provider.
description	Yes	String	Identity provider description.
enabled	Yes	Boolean	Whether an identity provider is enabled.
remote_ids	Yes	Array	Federated user ID list of an identity provider.

Parameter	Mandatory	Type	Description
links	Yes	Object	Identity provider resource link, including protocols and self .

- Example response

```
{
  "identity_provider": {
    "description": "Stores ACME identities",
    "enabled": true,
    "id": "ACME",

    "remote_ids": [],
    "links": {
      "protocols": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME/protocols",
      "self": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME"
    }
  }
}
```

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.
405	The method specified in the request is not allowed for the requested resource.
409	Duplicate identity provider ID.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.2.4 Updating a SAML Identity Provider

Function

This API is used to update the information about a SAML identity provider.

URI

- URI format
PATCH /v3/OS-FEDERATION/identity_providers/{id}
- URI parameters

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an identity provider.

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
identity_provider	Yes	JSON object	Request body for updating an identity provider. The request body must contain at least one parameter.
description	No	String	Identity provider description.
enabled	No	Boolean	Whether an identity provider is enabled. <ul style="list-style-type: none"> • true indicates that the identity provider is enabled. • false indicates that the identity provider is disabled. The default value is false .

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H 'X-Auth-Token:$token' -X PATCH -d '{"identity_provider":{"enabled":false}}' https://sample.domain.com/v3/OS-FEDERATION/identity_providers/ACME
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an identity provider.
description	Yes	String	Identity provider description.
enabled	Yes	Boolean	Whether an identity provider is enabled.
remote_ids	Yes	Array	Federated user ID list of an identity provider.
links	Yes	Object	Identity provider resource link.

- Example response

```
{
  "identity_provider": {
    "description": "Stores ACME identities",
    "enabled": false,
    "id": "ACME",

    "remote_ids": [],
    "links": {
      "protocols": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME/protocols",
      "self": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME"
    }
  }
}
```

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.
405	The method specified in the request is not allowed for the requested resource.
409	A resource conflict occurs.
413	The request entity is too large.
500	Internal server error.

Status Code	Description
503	Service unavailable.

4.12.2.5 Deleting an Identity Provider

Function

This API is used to delete a SAML or OpenID Connect identity provider.

URI

- URI format
DELETE /v3/OS-FEDERATION/identity_providers/{id}
- URI parameters

Parameter	Mandatory	Type	Description
id	Yes	String	ID of an identity provider.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H 'X-Auth-Token:$token' -X DELETE https://sample.domain.com/v3/OS-FEDERATION/identity_providers/ACME
```

Response Parameters

None

Status Codes

Status Code	Description
204	The request is successful.
400	The server failed to process the request.

Status Code	Description
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.3 Mapping

4.12.3.1 Querying the Mapping List

Function

This API is used to query the mapping list.

URI

GET /v3/OS-FEDERATION/mappings

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.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H 'X-Auth-Token:$token' -X GET https://sample.domain.com/v3/OS-FEDERATION/mappings
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
mappings	Yes	Array	List of mappings.
links	Yes	Object	Mapping resource link.

- mappings parameter description

Parameter	Mandatory	Type	Description
id	Yes	String	Mapping ID.

Parameter	Mandatory	Type	Description
rules	Yes	Object	<p>Rule used to map federated users to local users</p> <p>Example rule for SAML:</p> <pre> "rules": [{ "local": [{ "user": { "name": "{0}" } }, { "group": { "name": "0cd5e9" } }], "remote": [{ "type": "UserName" }, { "type": "orgPersonType", "not_any_of": ["Contractor", "Guest"] }] }] </pre> <p>local: indicates the information about a federated user in the cloud system.</p> <ul style="list-style-type: none"> ● user: indicates the name of a federated user in the cloud system. {0} indicates the first attribute of the user information in remote. ● group: indicates the user group to which a federated user belongs in the cloud system. <p>remote: indicates the information about a federated user in the IdP. This expression is a combination of assertion attributes and operators. The value of remote is determined based on the assertion.</p> <ul style="list-style-type: none"> ● "type": "UserName" indicates an attribute in an IdP assertion.

Parameter	Mandatory	Type	Description
			<ul style="list-style-type: none"> • "type": "orgPersonType" indicates an attribute in an IdP assertion. • not_any_of: The rule is not matched if any of the specified strings appear in the attribute type. The condition result is Boolean, not the argument that is passed as input.
links	Yes	Object	Mapping resource link.

- Example response

```

{
  "links": {
    "next": null,
    "previous": null,
    "self": "https://example.com/v3/OS-FEDERATION/mappings"
  },
  "mappings": [
    {
      "id": "ACME",
      "links": {
        "self": "https://example.com/v3/OS-FEDERATION/mappings/ACME"
      },
      "rules": [
        {
          "local": [
            {
              "user": {
                "name": "{0}"
              }
            },
            {
              "group": {
                "id": "0cd5e9"
              }
            }
          ],
          "remote": [
            {
              "type": "UserName"
            },
            {
              "type": "orgPersonType",
              "any_one_of": [
                "Contractor",
                "SubContractor"
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.3.2 Querying a Mapping

Function

This API is used to query the information about a mapping.

URI

- URI format
GET /v3/OS-FEDERATION/mappings/{id}
- URI parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Mapping ID.

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.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/OS-FEDERATION/mappings/ACME
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
id	Yes	String	Mapping ID.

Parameter	Mandatory	Type	Description
rules	Yes	Object	<p>Rule used to map federated users to local users</p> <p>Example rule for SAML:</p> <pre data-bbox="1026 405 1428 1144"> "rules": [{ "local": [{ "user": { "name": "{0}" } }, { "group": { "name": "0cd5e9" } }], "remote": [{ "type": "UserName" }, { "type": "orgPersonType", "not_any_of": ["Contractor", "Guest"] }] }] </pre> <p>local: indicates the information about a federated user in the cloud system.</p> <ul data-bbox="1026 1272 1428 1579" style="list-style-type: none"> ● user: indicates the name of a federated user in the cloud system. {0} indicates the first attribute of the user information in remote. ● group: indicates the user group to which a federated user belongs in the cloud system. <p>remote: indicates the information about a federated user in the IdP. This expression is a combination of assertion attributes and operators. The value of remote is determined based on the assertion.</p> <ul data-bbox="1026 1839 1428 1937" style="list-style-type: none"> ● "type": "UserName" indicates an attribute in an IdP assertion.

Parameter	Mandatory	Type	Description
			<ul style="list-style-type: none"> • "type": "orgPersonType" indicates an attribute in an IdP assertion. • not_any_of: The rule is not matched if any of the specified strings appear in the attribute type. The condition result is Boolean, not the argument that is passed as input.
links	Yes	Object	Mapping resource link.

• Example response

```

{
  "mapping": {
    "id": "ACME",
    "links": {
      "self": "https://example.com/v3/OS-FEDERATION/mappings/ACME"
    },
    "rules": [
      {
        "local": [
          {
            "user": {
              "name": "{0}"
            }
          },
          {
            "group": {
              "name": "0cd5e9"
            }
          }
        ],
        "remote": [
          {
            "type": "UserName"
          },
          {
            "type": "orgPersonType",
            "not_any_of": [
              "Contractor",
              "Guest"
            ]
          }
        ]
      }
    ]
  }
}

```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.

Status Code	Description
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.3.3 Creating a Mapping

Function

This API is used to create a mapping.

URI

- URI format
PUT /v3/OS-FEDERATION/mappings/{id}
- URI parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Mapping ID.

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
rules	Yes	Object	<p>Rule used to map federated users to local users.</p> <p>Example rule for SAML:</p> <pre data-bbox="1010 443 1428 1182"> "rules": [{ "local": [{ "user": { "name": "{0}" } }, { "group": { "name": "0cd5e9" } }], "remote": [{ "type": "UserName" }, { "type": "orgPersonType", "not_any_of": ["Contractor", "Guest"] }] }] </pre> <p>local: indicates the information about a federated user in the cloud system.</p> <ul data-bbox="1010 1305 1428 1615" style="list-style-type: none"> • user: indicates the name of a federated user in the cloud system. {0} indicates the first attribute of the user information in remote. • group: indicates the user group to which a federated user belongs in the cloud system. <p>remote: indicates the information about a federated user in the IdP. This expression is a combination of assertion attributes and operators. The value of remote is determined based on the assertion.</p> <ul data-bbox="1010 1877 1428 1966" style="list-style-type: none"> • "type": "UserName" indicates an attribute in an IdP assertion.

Parameter	Mandatory	Type	Description
			<ul style="list-style-type: none"> "type": "orgPersonType" indicates an attribute in an IdP assertion. not_any_of: The rule is not matched if any of the specified strings appear in the attribute type. The condition result is Boolean, not the argument that is passed as input.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X PUT -d '{"mapping":{"rules":[{"local":{"user":{"name":"{0}"},{"group":{"name":"Ocd5e9"}}, {"remote":[{"type":"UserName"}, {"type":"orgPersonType", "not_any_of":["Contractor","Guest"]}]}]}}' https://sample.domain.com/v3/OS-FEDERATION/mappings/ACME
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
id	Yes	String	Mapping ID.

Parameter	Mandatory	Type	Description
rules	Yes	Object	<p>Rule used to map federated users to local users.</p> <p>Example rule for SAML:</p> <pre> "rules": [{ "local": [{ "user": { "name": "{0}" } }, { "group": { "name": "0cd5e9" } }], "remote": [{ "type": "UserName" }, { "type": "orgPersonType", "not_any_of": ["Contractor", "Guest"] }] }] </pre> <p>local: indicates the information about a federated user in the cloud system.</p> <ul style="list-style-type: none"> • user: indicates the name of a federated user in the cloud system. {0} indicates the first attribute of the user information in remote. • group: indicates the user group to which a federated user belongs in the cloud system. <p>remote: indicates the information about a federated user in the IdP. This expression is a combination of assertion attributes and operators. The value of remote is determined based on the assertion.</p> <ul style="list-style-type: none"> • "type": "UserName" indicates an attribute in an IdP assertion.

Parameter	Mandatory	Type	Description
			<ul style="list-style-type: none"> • "type": "orgPersonType" indicates an attribute in an IdP assertion. • not_any_of: The rule is not matched if any of the specified strings appear in the attribute type. The condition result is Boolean, not the argument that is passed as input.
links	Yes	Object	Mapping resource link.

- Example response

```

{
  "mapping": {
    "id": "ACME",
    "links": {
      "self": "https://example.com/v3/OS-FEDERATION/mappings/ACME"
    },
    "rules": [
      {
        "local": [
          {
            "user": {
              "name": "{0}"
            }
          },
          {
            "group": {
              "name": "0cd5e9"
            }
          }
        ],
        "remote": [
          {
            "type": "UserName"
          },
          {
            "type": "orgPersonType",
            "not_any_of": [
              "Contractor",
              "Guest"
            ]
          }
        ]
      }
    ]
  }
}

```

Status Codes

Status Code	Description
201	The request is successful.
400	The server failed to process the request.

Status Code	Description
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
409	A resource conflict occurs.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.3.4 Updating a Mapping

Function

This API is used to update the information about a mapping.

URI

- URI format
PATCH /v3/OS-FEDERATION/mappings/{id}
- URI parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Mapping ID.

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
rules	Yes	Object	<p>Rule used to map federated users to local users.</p> <p>Example rule for SAML:</p> <pre data-bbox="1002 443 1430 1178"> "rules": [{ "local": [{ "user": { "name": "{0}" } }, { "group": { "name": "0cd5e9" } }], "remote": [{ "type": "UserName" }, { "type": "orgPersonType", "not_any_of": ["Contractor", "Guest"] }] }] </pre> <p>local: indicates the information about a federated user in the cloud system.</p> <ul data-bbox="1002 1305 1414 1615" style="list-style-type: none"> • user: indicates the name of a federated user in the cloud system. {0} indicates the first attribute of the user information in remote. • group: indicates the user group to which a federated user belongs in the cloud system. <p>remote: indicates the information about a federated user in the IdP. This expression is a combination of assertion attributes and operators. The value of remote is determined based on the assertion.</p> <ul data-bbox="1002 1877 1385 1966" style="list-style-type: none"> • "type": "UserName" indicates an attribute in an IdP assertion.

Parameter	Mandator y	Type	Description
			<ul style="list-style-type: none"> • "type": "orgPersonType" indicates an attribute in an IdP assertion. • not_any_of: The rule is not matched if any of the specified strings appear in the attribute type. The condition result is Boolean, not the argument that is passed as input.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X PATCH -d '{"mapping":{"rules":[{"local":[{"user":{"name":"","group":{"name":"Ocd5e9"}}, "remote":[{"type":"UserName"}, {"type":"orgPersonType", "any_one_of":["Contractor", "SubContractor"]}]}]}}' https://sample.domain.com/v3/OS-FEDERATION/mappings/ACME
```

Response Parameters

- Parameters in the response body

Parameter	Mandator y	Type	Description
id	Yes	String	Mapping ID.

Parameter	Mandatory	Type	Description
rules	Yes	Object	<p>Rule used to map federated users to local users.</p> <p>Example rule for SAML:</p> <pre data-bbox="997 443 1428 1176"> "rules": [{ "local": [{ "user": { "name": "{0}" } }, { "group": { "name": "0cd5e9" } }], "remote": [{ "type": "UserName" }, { "type": "orgPersonType", "not_any_of": ["Contractor", "Guest"] }] }] </pre> <p>local: indicates the information about a federated user in the cloud system.</p> <ul data-bbox="997 1305 1412 1615" style="list-style-type: none"> • user: indicates the name of a federated user in the cloud system. {0} indicates the first attribute of the user information in remote. • group: indicates the user group to which a federated user belongs in the cloud system. <p>remote: indicates the information about a federated user in the IdP. This expression is a combination of assertion attributes and operators. The value of remote is determined based on the assertion.</p> <ul data-bbox="997 1877 1380 1966" style="list-style-type: none"> • "type": "UserName" indicates an attribute in an IdP assertion.

Parameter	Mandatory	Type	Description
			<ul style="list-style-type: none"> "type": "orgPersonType" indicates an attribute in an IdP assertion. not_any_of: The rule is not matched if any of the specified strings appear in the attribute type. The condition result is Boolean, not the argument that is passed as input.
links	Yes	Object	Mapping resource link.

- Example response

```

{
  "mapping": {
    "id": "ACME",
    "links": {
      "self": "https://example.com/v3/OS-FEDERATION/mappings/ACME"
    }
  },
  "rules": [
    {
      "local": [
        {
          "user": {
            "name": "{0}"
          }
        },
        {
          "group": {
            "name": "0cd5e9"
          }
        }
      ],
      "remote": [
        {
          "type": "UserName"
        },
        {
          "type": "orgPersonType",
          "any_one_of": [
            "Contractor",
            "SubContractor"
          ]
        }
      ]
    }
  ]
}

```

Status Codes

Status Code	Description
200	The request is successful.

Status Code	Description
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
409	A resource conflict occurs.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.3.5 Deleting a Mapping

Function

This API is used to delete the information about a mapping.

URI

- URI format
DELETE /v3/OS-FEDERATION/mappings/{id}
- URI parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Mapping ID.

Request Parameters

- Parameters in the request header

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

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated token with the Security Administrator permission.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X DELETE https://sample.domain.com/v3/OS-FEDERATION/mappings/ACME
```

Response Parameters

None

Status Codes

Status Code	Description
204	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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.4 Protocol

4.12.4.1 Querying the Protocol List

Function

This API is used to query the protocol list.

URI

- URI format
GET /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols
- URI parameters

Parameter	Mandatory	Type	Description
idp_id	Yes	String	ID of an identity provider.

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.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H 'X-Auth-Token:$token' -X GET https://sample.domain.com/v3/OS-FEDERATION/identity_providers/ACME/protocols/
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
protocols	Yes	List of objects	List of protocols.
links	Yes	Object	Protocol resource link.

- protocols parameter description

Parameter	Mandatory	Type	Description
id	Yes	String	ID of a protocol.
mapping_id	Yes	String	Mapping ID.
links	Yes	Object	Protocol resource link.

- Example response

```
{
  "links": {
    "next": null,
    "previous": null,
  }
}
```

```

    "self": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME/protocols"
  },
  "protocols": [
    {
      "id": "saml",
      "links": {
        "identity_provider": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME",
        "self": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME/protocols/saml"
      },
      "mapping_id": "ACME"
    }
  ]
}

```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.4.2 Querying a Protocol

Function

This API is used to query the information about a protocol.

URI

- URI format
GET /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}
- URI parameters

Parameter	Mandatory	Type	Description
idp_id	Yes	String	ID of an identity provider.

Parameter	Mandatory	Type	Description
protocol_id	Yes	String	ID of a protocol.

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.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H 'X-Auth-Token:$token' -X GET https://sample.domain.com/v3/OS-FEDERATION/identity_providers/ACME/protocols/saml
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
id	Yes	String	ID of a protocol.
mapping_id	Yes	String	Mapping ID.
links	Yes	Object	Protocol resource link.

- Example response

```
{
  "protocol": {
    "id": "saml",
    "links": {
      "identity_provider": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME",
      "self": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME/protocols/saml"
    }
  },
  "mapping_id": "ACME"
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.

Status Code	Description
403	Access denied.
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.4.3 Registering a Protocol

Function

This API is used to register a protocol, that is, associate a rule with an identity provider.

URI

- URI format
PUT /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}
- URI parameters

Parameter	Mandatory	Type	Description
idp_id	Yes	String	ID of an identity provider.
protocol_id	Yes	String	ID of a protocol.

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
mapping_id	Yes	String	Mapping ID.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H 'X-Auth-Token:$token' -X PUT -d '{"protocol":{"mapping_id":"ACME"}}' https://sample.domain.com/v3/OS-FEDERATION/identity_providers/ACME/protocols/saml
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
id	Yes	String	ID of a protocol.
mapping_id	Yes	String	Mapping ID.
links	Yes	Object	Protocol resource link.

- Example response

```
{
  "protocol": {
    "id": "saml",
    "links": {
      "identity_provider": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME",
      "self": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME/protocols/saml"
    }
  },
  "mapping_id": "ACME"
}
```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.4.4 Updating a Protocol

Function

This API is used to update the information about a protocol.

URI

- URI format
PATCH /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}
- URI parameters

Parameter	Mandatory	Type	Description
idp_id	Yes	String	ID of an identity provider.
protocol_id	Yes	String	ID of a protocol.

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
mapping_id	Yes	String	Mapping ID.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X PATCH -d '{"protocol":{"mapping_id":"ACME"}}' https://sample.domain.com/v3/OS-FEDERATION/identity_providers/ACME/protocols/saml
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
id	Yes	String	ID of a protocol.
mapping_id	Yes	String	Mapping ID.
links	Yes	Object	Protocol resource link.

- Example response

```
{
  "protocol": {
    "id": "saml",
    "links": {
      "identity_provider": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME",
      "self": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME/protocols/saml"
    }
  },
  "mapping_id": "ACME"
}
```

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.
405	The method specified in the request is not allowed for the requested resource.
409	A resource conflict occurs.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.4.5 Deleting a Protocol

Function

This API is used to delete the information about a protocol.

URI

- URI format
DELETE /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}
- URI parameters

Parameter	Mandatory	Type	Description
idp_id	Yes	String	ID of an identity provider.
protocol_id	Yes	String	ID of a protocol.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H "X-Auth-Token:$token" -X DELETE https://sample.domain.com/v3/OS-FEDERATION/identity_providers/ACME/protocols/saml
```

Response Parameters

None

Status Codes

Status Code	Description
204	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.
405	The method specified in the request is not allowed for the requested resource.

Status Code	Description
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.5 Metadata

4.12.5.1 Querying a Metadata File

Function

This API is used to query the content of the metadata file imported by an identity provider to the IAM system.

URI

- URI format
GET /v3-ext/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}/metadata
- URI parameters

Parameter	Mandatory	Type	Description
idp_id	Yes	String	ID of an identity provider.
protocol_id	Yes	String	ID of a protocol.

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 with the Security Administrator permission.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H "X-Auth-Token: $token" -X GET https://sample.domain.com/v3-ext/OS-FEDERATION/identity_providers/ACME/protocols/saml/metadata
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
id	Yes	String	ID of a metadata file.
idp_id	Yes	String	ID of an identity provider.
entity_id	Yes	String	entityID field in the metadata file.
protocol_id	Yes	String	ID of a protocol.
domain_id	Yes	String	ID of the domain that a user belongs to.
xaccount_type	Yes	String	Domain source. The value is left empty by default.
update_time	Yes	String	Time when a metadata file is imported or updated.
data	Yes	String	Content of a metadata file.

- Example response

```
{
  "id": "40c174f35ff94e31b8257ad4991bce8b",
  "idp_id": "ACME",
  "entity_id": "https://idp.test.com/idp/shibboleth",
  "protocol_id": "saml",
  "domain_id": "ed7a77d365304f458f7d0a7909c6d889",
  "xaccount_type": "",
  "update_time": "2016-10-26T09:26:23.000000",
  "data": "$data"}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
500	Internal server error.

4.12.5.2 Querying the Metadata File of Keystone

Function

This API is used to query the metadata file of the keystone.

URI

GET /v3-ext/auth/OS-FEDERATION/SSO/metadata

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
unsigned	No	Boolean	Whether to sign metadata according to SAML 2.0 specifications. The default value of this parameter is false .

- Example request

GET /v3-ext/auth/OS-FEDERATION/SSO/metadata

Response Parameters

Example response

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
ID="43ebac773925f6849b196a3c803baba5" entityID="https://www.example.com">
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
<ds:Reference URI="#43ebac773925f6849b196a3c803baba5">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<ds:DigestValue>yuQJc6O13xilt6X4cOEUBnVV2Vs=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>...</ds:SignatureValue>
<ds:KeyInfo>
<ds:X509Data>
<ds:X509Certificate>...</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<md:SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<md:KeyDescriptor use="signing">
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:X509Data>
<ds:X509Certificate>...</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</md:KeyDescriptor>
<md:KeyDescriptor use="encryption">
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
```

```
<ds:X509Data>
<ds:X509Certificate>...</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</md:KeyDescriptor>
<md:NameIDFormat xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
urn:oasis:names:tc:SAML:2.0:nameid-format:transient
</md:NameIDFormat>
<md:AssertionConsumerService xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://www.example.com/v3-ext/
auth/OS-FEDERATION/SSO/SAML2/POST" index="0" isDefault="true"/>
<md:AssertionConsumerService xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS" Location="https://www.example.com/v3-ext/auth/OS-
FEDERATION/SSO/SAML2/ECP" index="1"/>
</md:SPSSODescriptor>
</md:EntityDescriptor>
```

Status Code

Status Code	Description
200	The request is successful.
500	Internal server error.
503	Service unavailable.

4.12.5.3 Importing a Metadata File

Function

Before using the federated identity authentication function, a metadata file must be imported to the IAM system. This API is used to import a metadata file of a domain.

URI

- URI format
POST /v3-ext/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}/metadata
- URI parameters

Parameter	Mandatory	Type	Description
idp_id	Yes	String	Identity provider ID.
protocol_id	Yes	String	Protocol ID.

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 with the Security Administrator permission.

- Parameters in the request body

Parameter	Mandatory	Type	Description
xaccount_type	Yes	String	Source of a domain. This field is left blank by default.
metadata	Yes	String	Content of the metadata file on the IdP server.
domain_id	Yes	String	ID of the domain that a user belongs to.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H 'X-Auth-Token:$token' -X POST -d '{"xaccount_type":"","domain_id":"ed7a77d365304f458f7d0a7909c6d889","metadata":"$metadataContent"}' https://sample.domain.com/v3-ext/OS-FEDERATION/identity_providers/ACME/protocols/saml/metadata
```

Response Parameters

Example response

```
{"message": "Import metadata successful"}
```

Status Codes

Status Code	Description
201	The import is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
500	Internal server error.

4.12.6 Token

4.12.6.1 Obtaining an Unscoped Token (SP Initiated)

Function

This API is used to obtain an unscoped token in SP-initiated federated identity authentication mode.

An unscoped token cannot be used for authentication. If a federated user needs to use a token for authentication, obtain the scoped token based on section [4.12.6.3 Obtaining a Scoped Token](#).

URI

- URI format
GET /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}/auth
- URI parameters

Parameter	Mandatory	Type	Description
idp_id	Yes	String	ID of an identity provider.
protocol_id	Yes	String	ID of a protocol.

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
Accept	No	String	<ul style="list-style-type: none"> • This parameter is not required when a token is obtained in the WebSSO mode. • When you obtain a token using the Enhanced Client Proxy (ECP), the value of this parameter is as follows: application/vnd.paos+xml
PAOS	No	String	<ul style="list-style-type: none"> • This parameter is not required when a token is obtained in the WebSSO mode. • When you obtain a token using the ECP, the value of this parameter is as follows: urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp

 **NOTE**

1. This API can be used to obtain tokens through WebSSO and ECP. Different request headers are used to determine the method of obtaining a token. For details, see the parameter description of Request Header.
2. You are not advised to obtain a token by directly calling this API. You are advised to obtain a token using OpenStackClient.

- Example request

```
GET /v3/OS-FEDERATION/identity_providers/idptest/protocols/saml/auth
```

Response Parameters

- Parameters in the response body

Response Item	Parameter	Type	Description
X-Subject-Token	header	String	Signed unscoped token.
token	body	Object	Information of the unscoped token obtained in federated identity authentication mode, including methods and user information.

- Example response

```
{
  "token": {
    "issued_at": "2017-05-23T06:54:51.763000Z",
    "expires_at": "2017-05-24T06:54:51.763000Z",
    "methods": [
      "mapped"
    ],
    "user": {
      "domain": {
        "id": "e31ac82d778b4d128cb6fed37fd72cdb",
        "name": "exampledomain"
      },
      "id": "RMQTgtjjSNGDcKy7oUml3AZg7GgsWG0Z",
      "name": "exampleuser",
      "OS-FEDERATION": {
        "identity_provider": {
          "id": "exampleuser"
        },
        "protocol": {
          "id": "saml"
        }
      },
      "groups": [
        {
          "id": "b40189e26ea44f959877621b4b298db5"
        }
      ]
    }
  }
}
```

Status Code

Status Code	Description
200	The request is successful. You need to further obtain user information.
201	The request is successful, and a token is returned.
302	The system switches to the identity provider authentication page if the request does not carry user information of the identity provider.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.6.2 Obtaining an Unscoped Token (IdP Initiated)

Function

This API is used to obtain an unscoped token in IdP-initiated federated identity authentication mode.

An unscoped token cannot be used for authentication. If a federated user needs to use a token for authentication, obtain the scoped token based on section [4.12.6.3 Obtaining a Scoped Token](#).

URI

POST /v3.0/OS-FEDERATION/tokens

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Idp-Id	Yes	String	ID of an identity provider.
Content-Type	Yes	String	The client must transfer the SAMLResponse parameter to the server by using the form data submitted by the browser. Therefore, the value of this parameter must be: application/x-www-form-urlencoded

- Parameters in the request body

Parameter	Mandatory	Type	Description
SAMLResponse	Yes	String	Response body returned when IdP authentication is successful.

NOTE

This API can only be called on the CLI side. The client needs to obtain SAMLResponse in IdP-initiated federated identity authentication mode and obtain an unscoped token by using the form data submitted by the browser.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'x-Idp-Id:test_local_idp' -H 'Content-Type:application/x-www-form-urlencoded' -X POST -d 'SAMLResponse=PD94bWwgdmVyc2lvbj0iMS4wL2ltZGU4LWxz1OGNmYmRzWk1ZeWlK96anpEbm1rT2FrVjBrUmIWEpLYUt5NzJtUmt0RFBCNjgwVQpzaU3R2hKNHE4ZG48L3h1bM6Q2lwaGVyVmFsdWU%2BPC94ZW5jOkNpcGhlckRhdGE%2BPC94ZW5jOkVvY3J5cHRIZERhdGE%2BPC9zYW1sMjFmNmNyeXB0ZWRCc3NlcnRpb24%2BPC9zYW1sMnA6UmVzcG9uc2U%2B' https://sample.domain.com/v3.0/OS-FEDERATION/tokens
```

Response Parameters

- Parameters in the response body

Response Item	Parameter	Type	Description
X-Subject-Token	header	String	Signed unscoped token.
token	body	Object	Information of the unscoped token obtained in federated identity authentication mode, including methods and user information.

- Example response

```
{
  "token": {
    "expires_at": "2018-03-13T03:00:01.168000Z",
    "methods": ["mapped"],
    "issued_at": "2018-03-12T03:00:01.168000Z",
    "user": {
      "OS-FEDERATION": {
        "identity_provider": {
          "id": "test_local_idp"
        },
        "protocol": {
          "id": "saml"
        },
        "groups": [{
          "name": "admin",
          "id": "45a8c8f1894444e9a016af065e152b91"
        }]
      },
      "domain": {
        "name": "hansheng",
        "id": "c0e20cc993a24ad4aa3251661ef37c87"
      },
      "name": "FederationUser",
      "id": "QNSzD0bycqUXE4hiRNfyFcWfoOs8z6gT"
    }
  }
}
```

Status Code

Status Code	Description
201	The request is successful, and a token is returned.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.12.6.3 Obtaining a Scoped Token

Function

This API is used to obtain a scoped token through federated identity authentication.

URI

POST /v3/auth/tokens

Request Parameters

Table 4-239 Parameters in the request header

Parameter	Mandatory	Type	Description
Content-Type	No	String	Fill application/json; charset=utf8 in this field.

Table 4-240 Parameters in the request body

Parameter	Mandatory	Type	Description
auth	Yes	Object	Authentication information.

Table 4-241 auth

Parameter	Mandatory	Type	Description
identity	Yes	Object	Authentication parameters.
scope	Yes	Object	Application scope of the token. The value can be project or domain .

Table 4-242 auth.identity

Parameter	Mandatory	Type	Description
methods	Yes	Array of strings	Authentication method. The value of this field is token .
token	Yes	Object	Unscoped token information.

Table 4-243 auth.identity.token

Parameter	Mandatory	Type	Description
id	Yes	String	Unscoped token ID.

Table 4-244 auth.scope

Parameter	Mandatory	Type	Description
domain	No	Object	If this field is set to domain , the token can be used to access resources in all projects under the account of a specified ID or name.
project	No	Object	If this field is set to project , the token can only be used to access resources in the project of a specified ID or name.

Table 4-245 auth.scope.domain

Parameter	Mandatory	Type	Description
id	No	String	Domain ID. Either id or name must be specified.
name	No	String	Domain name. Either id or name must be specified.

Table 4-246 auth.scope.project

Parameter	Mandatory	Type	Description
domain	No	Object	Domain information. This parameter is mandatory if the name parameter is set.
id	No	String	Project ID. Either id or name must be specified.
name	No	String	Project name. Either id or name must be specified.

Table 4-247 auth.scope.project.domain

Parameter	Mandatory	Type	Description
id	No	string	Domain ID. Either id or name must be specified.
name	No	string	Domain name. Either id or name must be specified.

Response Parameters

Table 4-248 Parameters in the response header

Parameter	Type	Description
X-Subject-Token	string	Signed scoped token.

Table 4-249 Parameters in the response body

Parameter	Type	Description
token	Object	Details of the scoped token.

Table 4-250 token

Parameter	Type	Description
methods	Array of strings	Method for obtaining the token.
expires_at	String	Time when the token will expire.
catalog	Array of objects	Catalog information.
domain	Object	Domain information of the IAM user who requests for the token. This parameter is returned only when the scope parameter in the request body has been set to domain .
project	Object	Project information of the user. This parameter is returned only when the scope parameter in the request body has been set to project .

Parameter	Type	Description
roles	Array of objects	Permissions information of the token.
user	Object	Information about the user who requests for the token.
issued_at	String	Time when the token was issued.

Table 4-251 token.catalog

Parameter	Type	Description
type	String	Type of the service to which the API belongs.
id	String	Service ID.
name	String	Service name.
endpoints	Array of objects	Endpoint information.

Table 4-252 token.catalog.endpoints

Parameter	Type	Description
url	String	Endpoint URL.
region	String	Region to which the endpoint belongs.
region_id	String	Region ID.
interface	String	Visibility of the API. public indicates that the API is available for public access.
id	String	Endpoint ID.

Table 4-253 token.domain

Parameter	Type	Description
name	String	Domain name.
id	String	Domain ID.

Table 4-254 token.project

Parameter	Type	Description
name	String	Project name.
id	String	Project ID.
domain	Object	Domain information of the project.

Table 4-255 token.project.domain

Parameter	Type	Description
name	String	Domain name.
id	String	Domain ID.

Table 4-256 token.roles

Parameter	Type	Description
name	String	Permission name.
id	String	Permission ID. The default value is 0 , which does not correspond to any permission.

Table 4-257 token.user

Parameter	Type	Description
domain	Object	Information about the domain used to create the user.
OS-FEDERATION	Object	Federated identity authentication information.
id	String	User ID.
name	String	Username.
password_expires_at	String	UTC time when the password will expire. If this parameter is empty, it indicates that the password has unlimited validity.

Table 4-258 token.user.domain

Parameter	Type	Description
name	String	Domain name.
id	String	Domain ID.

Table 4-259 token.user.OS-FEDERATION

Parameter	Type	Description
groups	Array of objects	User group information.
identity_provider	Object	Identity provider information.
protocol	Object	Protocol information.

Table 4-260 token.user.OS-FEDERATION.groups

Parameter	Type	Description
id	String	User group ID.
name	String	User group name.

Table 4-261 token.user.OS-FEDERATION.identity_provider

Parameter	Type	Description
id	String	Identity provider ID.

Table 4-262 token.user.OS-FEDERATION.protocol

Parameter	Type	Description
id	String	Protocol ID.

Example Request

```
POST https://sample.domain.com/v3/auth/tokens
{
  "auth": {
    "identity": {
      "methods": [
        "token"
      ],

```

```

    "token": {
      "id": "MIlatAYJKoZlIhvcNAQcCollapTCCGqECAQExDTALB..."
    },
    "scope": {
      "domain": {
        "id": "063bb260a480cecc0f36c0086bb6c..."
      }
    }
  }
}

```

Example Response

Status code: 201

The scoped token is obtained successfully.

Parameters in the response header

X-Subject-Token:MIlatAYJKoZlIhvcNAQcCollapTCCGqECAQExDTALB...

Parameters in the response body

```

{
  "token": {
    "expires_at": "2020-02-13T14:21:34.042000Z",
    "methods": [
      "token"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "id": "d2983f677ce14f1e81cbb6a9345a1...",
            "interface": "public",
            "region": "**",
            "region_id": "**",
            "url": "https://sample.domain.com/v3"
          }
        ],
        "id": "fd631b3426cb40f0919091d5861d8...",
        "name": "keystone",
        "type": "identity"
      }
    ],
    "domain": {
      "id": "06aa2260a480cecc0f36c0086bb6cfe0",
      "name": "IAMDomain"
    },
    "roles": [
      {
        "id": "0",
        "name": "te_admin"
      },
      {
        "id": "0",
        "name": "secu_admin"
      }
    ],
    "issued_at": "2020-02-12T14:21:34.042000Z",
    "user": {
      "OS-FEDERATION": {
        "groups": [
          {
            "id": "06aa2260bb00cecc3f3ac0084a74038f",
            "name": "admin"
          }
        ],
        "identity_provider": {
          "id": "ACME"
        }
      }
    }
  }
}

```

```
    "protocol": {
      "id": "saml"
    },
  },
  "domain": {
    "id": "06aa2260a480cecc0f36c0086bb6cfe0",
    "name": "IAMDomain"
  },
  "id": "LdQTDSC7zmJVlic3yaCbLBXDxPAdDxLg",
  "name": "FederationUser",
  "password_expires_at": ""
}
}
```

Status Codes

Status Code	Description
201	The scoped token is obtained successfully.
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.

Error Codes

None

4.12.7 Domain

4.12.7.1 Querying the List of Domains Accessible to Federated Users

Function

This API is used to query the list of domains accessible to federated users.

URI

GET /v3/OS-FEDERATION/domains

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Unscoped token. For details, see 4.12.6.1 Obtaining an Unscoped Token (SP Initiated) .

- Example request
GET /v3/OS-FEDERATION/domains

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
domains	Yes	array	List of domains.
links	Yes	Object	Domain resource link.

- Example response

```
{
  "domains": [
    {
      "links": {
        "self": "https://sample.domain.com/v3/domains/e31ac82d778b4d128cb6fed37fd72cdb"
      },
      "description": null,
      "name": "exampledomain",
      "enabled": true,
      "id": "e31ac82d778b4d128cb6fed37fd72cdb"
    }
  ],
  "links": {
    "self": "https://sample.domain.com/v3/OS-FEDERATION/domains",
    "previous": null,
    "next": null
  }
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.

Status Code	Description
500	Internal server error.
503	Service unavailable.

4.12.8 Project

4.12.8.1 Querying the List of Projects Accessible to Federated Users

Function

This API is used to query the list of projects accessible to federated users. The project list is used to obtain the scoped token in federated identity authentication mode.

URI

GET /v3/OS-FEDERATION/projects

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Unscoped token. For details about how to obtain a token, see 4.12.6.1 Obtaining an Unscoped Token (SP Initiated) .

- Example request

```
GET /v3/OS-FEDERATION/projects
```

Response Parameters

- Parameters in the response body

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

- Example response

```
{
  "links": {
    "self": "https://sample.domain.com/v3/OS-FEDERATION/projects",
    "previous": null,
    "next": null
  }
}
```



```

},
"projects": [
  {
    "is_domain": false,
    "description": "",
    "links": {
      "self": "https://sample.domain.com/v3/projects/05cf683c351e43518618d9fa96a5efa9"
    },
    "enabled": true,
    "id": "05cf683c351e43518618d9fa96a5efa9",
    "parent_id": "e31ac82d778b4d128cb6fed37fd72cdb",
    "domain_id": "e31ac82d778b4d128cb6fed37fd72cdb",
    "name": "region_name"
  },
  {
    "is_domain": false,
    "description": "",
    "links": {
      "self": "https://sample.domain.com/v3/projects/32b56f108f87418e8219317beb0fff3c"
    },
    "enabled": true,
    "id": "32b56f108f87418e8219317beb0fff3c",
    "parent_id": "e31ac82d778b4d128cb6fed37fd72cdb",
    "domain_id": "e31ac82d778b4d128cb6fed37fd72cdb",
    "name": "MOS" //Default project name of OBS
  }
]
}

```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
401	Authentication failed.
403	Access denied.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.13 Version Information Management

4.13.1 Querying Keystone API Version Information

Function

This API is used to obtain the keystone API version information.

URI

GET /

Request Parameters

Example request

```
curl -i -k -X GET https://sample.domain.com/
```

Response Parameters

- Response parameter description

Parameter	Mandatory	Type	Description
versions	Yes	Object	Keystone API version information.
values	Yes	Array	Keystone API version list.

- Description for the values format

Parameter	Mandatory	Type	Description
status	Yes	String	Version status.
updated	Yes	String	Last version update time.
media-types	Yes	Array	Version-supported message format.
id	Yes	String	Version, for example, v3.0.
links	Yes	Array	Version resource link.

- Example response (successful response)

```
{
  "versions": {
    "values": [
      {
        "media-types": [
          {
            "type": "application/vnd.openstack.identity-v3+json",
            "base": "application/json"
          }
        ],
        "links": [
          {
            "rel": "self",
            "href": "https://sample.domain.com/v3/"
          }
        ],
        "id": "v3.6",
        "updated": "2016-04-04T00:00:00Z",
        "status": "stable"
      }
    ]
  }
}
```

Status Codes

Status Code	Description
300	The request is successful.
400	The server failed to process the request.
404	The requested resource cannot be found.
503	Service unavailable.

4.13.2 Querying Information About Keystone API Version 3.0

Function

This API is used to obtain the information about the keystone API version 3.0.

URI

GET /v3

Request Parameters

Example request

```
curl -i -k -X GET https://sample.domain.com/v3
```

Response Parameters

- Response parameter description

Parameter	Mandatory	Type	Description
version	Yes	Object	Keystone API version information.

- Description for the version format

Parameter	Mandatory	Type	Description
status	Yes	String	Version status.
updated	Yes	String	Last version update time.
media-types	Yes	Array	Version-supported message format.
id	Yes	String	Version, for example, v3.0.

Parameter	Mandatory	Type	Description
links	Yes	Array	Version resource link.

- Example response (successful response)

```
{
  "version": {
    "status": "stable",
    "updated": "2016-04-04T00:00:00Z",
    "media-types": [
      {
        "base": "application/json",
        "type": "application/vnd.openstack.identity-v3+json"
      }
    ],
    "id": "v3.6",
    "links": [
      {
        "href": "https://sample.domain.com/v3/",
        "rel": "self"
      }
    ]
  }
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.
404	The requested resource cannot be found.
503	Service unavailable.

4.14 Services and Endpoints

4.14.1 Querying Services

Function

This API is used to query the service list.

URI

- URI format
GET /v3/services{?type}
- URI parameters

Parameter	Mandatory	Type	Description
type	No	String	Service type. The value can be compute , ec2 , identity , image , network , or volume .

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.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/services?type=compute
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
links	Yes	Dict	Service resource link.
services	Yes	List	A list of services.

- Description for the services format

Parameter	Mandatory	Type	Description
description	No	String	Service description.
enabled	Yes	Boolean	Whether a service is available.
id	Yes	String	Service ID.
name	No	String	Service name.
type	Yes	String	Service type.
links	Yes	Dict	Service resource link.

- Example response (successful response)

```
{
  "services": [
    {
```

```

    "name": "compute5",
    "links": {
      "self": "https://sample.domain.com/v3/services/053d21d488d1463c818132d9d08fb617"
    },
    "enabled": true,
    "type": "compute",
    "id": "053d21d488d1463c818132d9d08fb617",
    "description": "Compute service 5"
  },
  {
    "name": "compute3",
    "links": {
      "self": "https://sample.domain.com/v3/services/c2474183dca7453bbd73123a0b78feae"
    },
    "enabled": true,
    "type": "compute",
    "id": "c2474183dca7453bbd73123a0b78feae",
    "description": "Compute service 3"
  },
  {
    "name": "compute2",
    "links": {
      "self": "https://sample.domain.com/v3/services/c7166694ebdd4616bd927737f7b12ca2"
    },
    "enabled": true,
    "type": "compute",
    "id": "c7166694ebdd4616bd927737f7b12ca2",
    "description": "Compute service 2"
  }
],
"links": {
  "self": "https://sample.domain.com/v3/services?type=compute",
  "previous": null,
  "next": null
}
}

```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.14.2 Querying Service Details

Function

This API is used to query service details.

URI

- URI format
GET /v3/services/{service_id}
- URI parameters

Parameter	Mandatory	Type	Description
service_id	Yes	String	Service ID.

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.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H 'X-Auth-Token:$token' -X GET https://sample.domain.com/v3/services/5a4ed456d228428c800ed2b67b4363a7
```

Response Parameters

Example response (successful response)

```
{
  "service": {
    "enabled": true,
    "type": "compute",
    "name": "nova",
    "links": {
      "self": "sample.domain.com/v3/services/5a4ed456d228428c800ed2b67b4363a7"
    }
  },
  "id": "5a4ed456d228428c800ed2b67b4363a7"
}
```

Status Codes

Status Code	Description
200	The request is successful.
400	The server failed to process the request.

Status Code	Description
401	Authentication failed.
403	Access denied.
404	The requested resource cannot be found.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.14.3 Querying Endpoints

Function

This API is used to query the list of terminal addresses and provides a service access entry.

URI

- URI format
GET /v3/endpoints{?interface, service_id}
- URI parameters

Parameter	Mandatory	Type	Description
interface	No	String	Plane to which an endpoint belongs. The value can be public , internal , or admin . <ul style="list-style-type: none"> • public: Users can view it on the public network interface. • internal: Users can view it on the internal network interface. • admin: The administrator can view it on the secure network interface.
service_id	No	String	Service ID.

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.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:$token" -X GET https://sample.domain.com/v3/endpoints?interface=public&service_id=43cbe5e77aaf4665bbb962062dc1fc9d
```

Response Parameters

- Parameters in the response body

Parameter	Mandatory	Type	Description
links	Yes	dict	Endpoint resource link.
endpoints	Yes	list	List of endpoints.

- Description for the endpoints format

Parameter	Mandatory	Type	Description
id	Yes	String	Endpoint ID.
url	Yes	String	Terminal endpoint URL.
region	Yes	String	Region to which an endpoint belongs.
region_id	Yes	String	ID of the region to which an endpoint belongs.
enabled	Yes	Boolean	Whether an endpoint is available.
interface	Yes	String	Plane to which an endpoint belongs.
service_id	Yes	String	ID of the service to which an endpoint belongs.
links	Yes	dict	Endpoint resource link.

- Example response (successful request)

```
{
  "endpoints": [
    {
      "region_id": null,
      "links": {
```

```

        "self": "https://sample.domain.com/v3/endpoints/162277d696f54cf592f19b569f85d158"
    },
    "url": "https://sample.domain.com/v3",
    "region": null,
    "enabled": true,
    "interface": "public",
    "service_id": "053d21d488d1463c818132d9d08fb617",
    "id": "162277d696f54cf592f19b569f85d158"
  }
],
"links": {
  "self": "https://sample.domain.com/v3/endpoints?
service_id=053d21d488d1463c818132d9d08fb617&interface=public",
  "previous": null,
  "next": null
}
}

```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.14.4 Querying Endpoint Details

Function

This API is used to query endpoint details.

URI

- URI format
GET /v3/endpoints/{endpoint_id}
- URI parameters

Parameter	Mandatory	Type	Description
endpoint_id	Yes	String	Endpoint ID.

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.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H 'X-Auth-Token:$token' -X GET https://sample.domain.com/v3/endpoints/62ea3602f8ee42b1825956473f5295a8
```

Response Parameters

Example response (successful request)

```
{
  "endpoint": {
    "region_id": "region_id",
    "links": {
      "self": "https://sample.domain.com/v3/endpoints/62ea3602f8ee42b1825956473f5295a8"
    },
  },
  "url": "https://sample.domain.com/v2/",
  "region": "region_name",
  "enabled": true,
  "interface": "public",
  "service_id": "5a4ed456d228428c800ed2b67b4363a7",
  "id": "62ea3602f8ee42b1825956473f5295a8"
}
```

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.

Status Code	Description
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

4.14.5 Querying the Service Catalog

Function

This API is used to query the service catalog corresponding to **X-Auth-Token** contained in the request.

URI

GET /v3/auth/catalog

Request Parameters

- Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Authenticated scoped token of a project.

- Example request

```
curl -i -k -H 'Accept:application/json' -H 'X-Auth-Token:$token' -H 'Content-Type:application/json;charset=utf8' -X GET https://sample.domain.com/v3/auth/catalog
```

Response Parameters

Example response (successful request)

```
{
  "catalog": [
    {
      "endpoints": [
        {
          "region_id": null,
          "url": "https://sample.domain.com/v2/c972a59e958e407e89b0c6d8e522df3b",
          "region": null,
          "interface": "public",
          "id": "04f0ee42038447f0a9c7b407028fd7b9"
        }
      ],
      "type": "compute",
      "id": "eb884e9f64b44dd0ac73cdc55d817286",
      "name": "nova"
    }
  ]
}
```

```

    }
  ],
  "links": {
    "self": "https://sample.domain.com/v3/auth/catalog"
  }
}

```

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.
405	The method specified in the request is not allowed for the requested resource.
413	The request entity is too large.
500	Internal server error.
503	Service unavailable.

5 Permissions Policies and Supported Actions

[5.1 Introduction](#)

[5.2 Action List](#)

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.

5.2 Action List

Token Management

Permission	API	Action	IAM Project	Enterprise Project
Obtaining an Agency Token	POST /v3/auth/tokens	iam:tokens:assume	-	-

Access Key Management

Permission	API	Action	IAM Project	Enterprise Project
Listing Permanent Access Keys	GET /v3.0/OS-CREDENTIAL/credentials	iam:credentials:listCredentials	-	-
Querying a Permanent Access Key	GET /v3.0/OS-CREDENTIAL/credentials/{access_key}	iam:credentials:getCredential	-	-
Creating a Permanent Access Key	POST /v3.0/OS-CREDENTIAL/credentials	iam:credentials:createCredential	-	-
Modifying a Permanent Access Key	PUT /v3.0/OS-CREDENTIAL/credentials/{access_key}	iam:credentials:updateCredential	-	-

Permission	API	Action	IAM Project	Enterprise Project
Deleting a Permanent Access Key	DELETE /v3.0/OS-CREDENTIAL/credentials/{access_key}	iam:credentials:deleteCredential	-	-

Virtual MFA Device Management

Permission	API	Action	IAM Project	Enterprise Project
Unbinding a Virtual MFA Device	×	iam:mfa:unbindMFADevice	-	-
Binding a Virtual MFA Device	×	iam:mfa:bindMFADevice	-	-
Creating a Virtual MFA Device	×	iam:mfa:createVirtualMFADevice	-	-
Deleting a Virtual MFA Device	×	iam:mfa:deleteVirtualMFADevice	-	-

Project Management

Permission	API	Action	IAM Project	Enterprise Project
Creating a Project	POST /v3/projects	iam:projects:createProject	-	-
Modifying Project Data	PATCH /v3/projects/{project_id}	iam:projects:updateProject	-	-
Changing Project Status	PUT /v3-ext/projects/{project_id}	iam:projects:updateProject	-	-
Querying the List of Projects Accessible to Users	GET /v3/users/{user_id}/projects	iam:projects:listProjectsForUser	-	-

Permission	API	Action	IAM Project	Enterprise Project
Deleting a Project	×	iam:projects:deleteProject	-	-

User Management

Permission	API	Action	IAM Project	Enterprise Project
Listing Users	GET /v3/users	iam:users:listUsers	-	-
Querying User Details	GET /v3/users/{user_id}	iam:users:getUser	-	-
Querying User Details (Recommended)	GET /v3.0/OS-USER/users/{user_id}	iam:users:getUser	-	-
Querying the User Group to Which a User Belongs	GET /v3/users/{user_id}/groups	iam:groups:listGroupsForUser	-	-
Querying Users in a User Group	GET /v3/groups/{group_id}/users	iam:users:listUsersForGroup	-	-
Creating a User	POST /v3/users	iam:users:createUser	-	-
Modifying User Information	PATCH /v3/users/{user_id}	iam:users:updateUser	-	-
Deleting a User	DELETE /v3/users/{user_id}	iam:users:deleteUser	-	-
Resetting a User's Password	×	iam:users:resetUserPassword	-	-
Configuring Login Protection	×	iam:users:setUserLoginProtect	-	-
Listing Users Who Have Access to a Specified Project	×	iam:users:listUsersForProject	-	-

Permission	API	Action	IAM Project	Enterprise Project
Deleting a User from a User Group	DELETE /v3/groups/{group_id}/users/{user_id}	iam:permissions:removeUserFromGroup	-	-

User Group Management

Permission	API	Action	IAM Project	Enterprise Project
Querying Users in a User Group	GET /v3/groups/{group_id}/users	iam:users:listUsersForGroup	-	-
Listing User Groups	GET /v3/groups{?domain_id,name}	iam:groups:listGroups	-	-
Querying User Group Details	GET /v3/groups/{group_id}	iam:groups:getGroup	-	-
Creating a User Group	POST /v3/groups	iam:groups:createGroup	-	-
Adding a User to a User Group	PUT /v3/groups/{group_id}/users/{user_id}	iam:permissions:addUserToGroup	-	-
Updating User Group Information	PATCH /v3/groups/{group_id}	iam:groups:updateGroup	-	-

Permission	API	Action	IAM Project	Enterprise Project
Deleting a User Group	DELETE /v3/groups/{group_id}	<ul style="list-style-type: none"> iam:groups:deleteGroup iam:permissions:removeUserFromGroup iam:permissions:revokeRoleFromGroup iam:permissions:revokeRoleFromGroupOnProject iam:permissions:revokeRoleFromGroupOnDomain 	-	-
Checking Whether a User Belongs to a Specified User Group	HEAD /v3/groups/{group_id}/users/{user_id}	iam:permissions:checkUserInGroup	-	-

Permissions Management

Permission	API	Action	IAM Project	Enterprise Project
Querying a Role List	GET /v3/roles	iam:roles:listRoles	-	-
Querying Role Details	GET /v3/roles/{role_id}	iam:roles:getRole	-	-
Querying Permissions of a User Group Under a Domain	GET /v3/domains/{domain_id}/groups/{group_id}/roles	iam:permissions:listRolesForGroupOnDomain	-	-

Permission	API	Action	IAM Project	Enterprise Project
Querying Permissions of a User Group Corresponding to a Project	GET /v3/projects/{project_id}/groups/{group_id}/roles	iam:permissions:listRolesForGroupOnProject	-	-
Granting Permissions to a User Group of a Domain	PUT /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	iam:permissions:grantRoleToGroupOnDomain	-	-
Granting Permissions to a User Group Corresponding to a Project	PUT /v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	iam:permissions:grantRoleToGroupOnProject	-	-
Removing Permissions of a User Group Corresponding to a Project	DELETE /v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	iam:permissions:revokeRoleFromGroupOnProject	-	-
Removing Permissions of a User Group of a Domain	DELETE /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	iam:permissions:revokeRoleFromGroupOnDomain	-	-
Querying Whether a User Group Under a Domain Has Specific Permissions	HEAD /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	iam:permissions:checkRoleForGroupOnDomain	-	-
Querying Whether a User Group Corresponding to a Project Has Specific Permissions	HEAD /v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	iam:permissions:checkRoleForGroupOnProject	-	-

Permission	API	Action	IAM Project	Enterprise Project
Granting Permissions to a User Group	PUT /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id} PUT /v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	iam:permissions:grantRoleToGroup	-	-
Querying the Permissions Granted to a User for a Specified Project	x	iam:permissions:listRolesForUserOnProject	-	-
Querying All Permissions of a User Group	x	iam:permissions:listRolesForGroup	-	-
Checking Whether a User Group Has Specified Permissions	x	iam:permissions:checkRoleForGroup	-	-
Removing Permissions of a User Group	x	iam:permissions:revokeRoleFromGroup	-	-

Custom Policy Management

Permission	API	Action	IAM Project	Enterprise Project
Listing Custom Policies	GET /v3.0/OS-ROLE/roles	iam:roles:listRoles	-	-
Querying Custom Policy Details	GET /v3.0/OS-ROLE/roles/{role_id}	iam:roles:getRole	-	-
Creating a Custom Policy	POST /v3.0/OS-ROLE/roles	iam:roles:createRole	-	-
Modifying a Custom Policy	PATCH /v3.0/OS-ROLE/roles/{role_id}	iam:roles:updateRole	-	-

Permission	API	Action	IAM Project	Enterprise Project
Deleting a Custom Policy	DELETE /v3.0/OS-ROLE/roles/{role_id}	iam:roles:deleteRole	-	-

Agency Management

Permission	API	Action	IAM Project	Enterprise Project
Creating an Agency	POST /v3.0/OS-AGENCY/agencies	iam:agencies:createAgency	-	-
Listing Agencies	GET /v3.0/OS-AGENCY/agencies	iam:agencies:listAgencies	-	-
Querying Agency Details	GET /v3.0/OS-AGENCY/agencies/{agency_id}	iam:agencies:getAgency	-	-
Modifying an Agency	PUT /v3.0/OS-AGENCY/agencies/{agency_id}	iam:agencies:updateAgency	-	-
Deleting an Agency	DELETE /v3.0/OS-AGENCY/agencies/{agency_id}	iam:agencies:deleteAgency	-	-
Granting Permissions to an Agency for a Project	PUT /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id}	iam:permissions:grantRoleToAgencyOnProject	-	-
Checking Whether an Agency Has the Specified Permissions on a Project	HEAD /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id}	iam:permissions:checkRoleForAgencyOnProject	-	-
Querying Permissions of an Agency for a Project	GET /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles	iam:permissions:listRolesForAgencyOnProject	-	-

Permission	API	Action	IAM Project	Enterprise Project
Removing Permissions of an Agency on a Project	DELETE /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id}	iam:permissions:revokeRoleFromAgencyOnProject	-	-
Granting Permissions to an Agency on a Domain	PUT /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}	iam:permissions:grantRoleToAgencyOnDomain	-	-
Checking Whether an Agency Has the Specified Permissions on a Domain	HEAD /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}	iam:permissions:checkRoleForAgencyOnDomain	-	-
Querying the List of Permissions of an Agency on a Domain	GET /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles	iam:permissions:listRolesForAgencyOnDomain	-	-
Removing Permissions of an Agency on a Domain	DELETE /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}	iam:permissions:revokeRoleFromAgencyOnDomain	-	-

Federated Identity Authentication Management

Permission	API	Action	IAM Project	Enterprise Project
Querying the Identity Provider List	GET /v3/OS-FEDERATION/identity_providers	iam:identityProviders:listIdentityProviders	-	-
Querying an Identity Provider	GET /v3/OS-FEDERATION/identity_providers/{id}	iam:identityProviders:getIdentityProvider	-	-
Creating an Identity Provider	PUT /v3/OS-FEDERATION/identity_providers/{id}	iam:identityProviders:createIdentityProvider	-	-

Permission	API	Action	IAM Project	Enterprise Project
Updating an Identity Provider	PATCH /v3/OS-FEDERATION/identity_providers/{id}	iam:identityProviders:updateIdentityProvider	-	-
Deleting an Identity Provider	DELETE /v3/OS-FEDERATION/identity_providers/{id}	iam:identityProviders:deleteIdentityProvider	-	-
Querying the Mapping List	GET /v3/OS-FEDERATION/mappings	iam:identityProviders:listMappings	-	-
Querying Mapping Details	GET /v3/OS-FEDERATION/mappings/{id}	iam:identityProviders:getMapping	-	-
Creating a Mapping	PUT /v3/OS-FEDERATION/mappings/{id}	iam:identityProviders:createMapping	-	-
Updating a Mapping	PATCH /v3/OS-FEDERATION/mappings/{id}	iam:identityProviders:updateMapping	-	-
Deleting a Mapping	DELETE /v3/OS-FEDERATION/mappings/{id}	iam:identityProviders:deleteMapping	-	-
Querying the Protocol List	GET /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols	iam:identityProviders:listProtocols	-	-
Querying a Protocol	GET /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}	iam:identityProviders:getProtocol	-	-
Registering a Protocol	PUT /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}	iam:identityProviders:createProtocol	-	-
Updating a Protocol	PATCH /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}	iam:identityProviders:updateProtocol	-	-

Permission	API	Action	IAM Project	Enterprise Project
Deleting a Protocol	DELETE /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}	iam:identityProviders:deleteProtocol	-	-
Querying a Metadata File	GET /v3-ext/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}/metadata	iam:identityProviders:getIDPMetadata	-	-
Importing a Metadata File	POST /v3-ext/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}/metadata	iam:identityProviders:createIDPMetadata	-	-

6 Appendix

[6.1 Status Codes](#)

[6.2 Error Codes](#)

[6.3 Obtaining User, Account, User Group, Project, and Agency Information](#)

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.

Status Code	Message Title	Description
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.
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.

Status Code	Message Title	Description
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.
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.

Status Code	Message Title	Description
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.
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.

Status Code	Message Title	Description
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.

Status Code	Error Code	Error Message	Description	Measure
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.
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.

Status Code	Error Code	Error Message	Description	Measure
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.
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.

Status Code	Error Code	Error Message	Description	Measure
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.
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.

Status Code	Error Code	Error Message	Description	Measure
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.
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.

Status Code	Error Code	Error Message	Description	Measure
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.
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.

Status Code	Error Code	Error Message	Description	Measure
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.
400	IAM.1032	The OCP NotAction cannot be 'allow'.	The notAction field cannot be allow for organization control policies (OCs).	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.

Status Code	Error Code	Error Message	Description	Measure
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.
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 1500 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.

Status Code	Error Code	Error Message	Description	Measure
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.
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.

Status Code	Error Code	Error Message	Description	Measure
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.
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.

Status Code	Error Code	Error Message	Description	Measure
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.
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.

Status Code	Error Code	Error Message	Description	Measure
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.
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 <i>API Reference</i> .
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 <i>API Reference</i> .
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.

Status Code	Error Code	Error Message	Description	Measure
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.
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 authentication failed.	Access token authentication 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.

Status Code	Error Code	Error Message	Description	Measure
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.
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.

Status Code	Error Code	Error Message	Description	Measure
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	Description
2022-08-15	This issue is the first official release.