

OpenStack

Identity API v2.0

Identity Admin API v2.0

Identity API v3.0

(February 21, 2017)



docs.openstack.org

Identity API v2.0

- [Extensions](#)
 - [Show extension details](#)
 - [List extensions](#)
- [Tokens and tenants](#)
 - [List tenants](#)
 - [Authenticate](#)
- [API versions](#)
 - [Show version details](#)
 - [List versions](#)
- [API Details](#)
 - [Overview](#)
 - [Paginated collections](#)
 - [Request and response formats](#)
- [Revocations](#)
 - [List Revoked Tokens \(v2\)](#)

Identity Admin API v2.0

- [Tenants](#)
 - [Show tenant details, by ID](#)
 - [List users on a tenant](#)
 - [List roles for user](#)
 - [List tenants \(admin endpoint\)](#)
 - [Show tenant details, by name](#)
 - [Delete tenant](#)
 - [Update tenant](#)
 - [Create tenant](#)
- [Tokens](#)
 - [List endpoints for token](#)
 - [Authenticate for admin API](#)
 - [Validate token](#)
 - [Validate token \(admin\)](#)
 - [Delete token](#)
- [Users](#)
 - [List user global roles](#)
 - [Create user \(admin endpoint\)](#)
 - [List users \(admin endpoint\)](#)
 - [Update user \(admin endpoint\)](#)
 - [Delete user \(admin endpoint\)](#)
 - [Show user details \(admin endpoint\)](#)
- [Endpoints](#)
 - [List endpoint templates](#)
 - [Create endpoint template](#)
 - [Delete endpoint template](#)
- [Versions](#)
 - [Get version details](#)
- [Certificates](#)
 - [Show CA Certificate \(v2\)](#)
 - [Show Signing Certificate \(v2\)](#)

Identity API v3.0

- [What's New in Version 3.8](#)
- [What's New in Version 3.7](#)
- [What's New in Version 3.6](#)
- [What's New in Version 3.5](#)
- [What's New in Version 3.4](#)
- [What's New in Version 3.3](#)
- [What's New in Version 3.2](#)
- [What's New in Version 3.1](#)
- [What's New in Version 3.0](#)
- [Authentication and token management](#)
 - [Password authentication with unscoped authorization](#)
 - [Password authentication with scoped authorization](#)
 - [Password authentication with explicit unscoped authorization](#)
 - [Token authentication with unscoped authorization](#)
 - [Token authentication with scoped authorization](#)
 - [Validate and show information for token](#)
 - [Check token](#)
 - [Revoke token](#)
 - [Get service catalog](#)
 - [Get available project scopes](#)
 - [Get available domain scopes](#)
- [Credentials](#)
 - [Create credential](#)
 - [List credentials](#)
 - [Show credential details](#)
 - [Update credential](#)
 - [Delete credential](#)
- [Domains](#)
 - [List domains](#)
 - [Create domain](#)
 - [Show domain details](#)
 - [Update domain](#)
 - [Delete domain](#)
- [Domain configuration](#)
 - [Show default configuration settings](#)
 - [Show default configuration for a group](#)
 - [Show default option for a group](#)
 - [Show domain group option configuration](#)
 - [Update domain group option configuration](#)
 - [Delete domain group option configuration](#)
 - [Show domain group configuration](#)
 - [Update domain group configuration](#)
 - [Delete domain group configuration](#)
 - [Create domain configuration](#)
 - [Show domain configuration](#)
 - [Update domain configuration](#)
 - [Delete domain configuration](#)
- [Groups](#)
 - [List groups](#)
 - [Create group](#)

- [Show group details](#)
- [Update group](#)
- [Delete group](#)
- [List users in group](#)
- [Add user to group](#)
- [Check whether user belongs to group](#)
- [Remove user from group](#)
- [OS-INHERIT API](#)
 - [Assign role to user on projects owned by domain](#)
 - [Assign role to group on projects owned by a domain](#)
 - [List user's inherited project roles on a domain](#)
 - [List group's inherited project roles on domain](#)
 - [Check if user has an inherited project role on domain](#)
 - [Check if group has an inherited project role on domain](#)
 - [Revoke an inherited project role from user on domain](#)
 - [Revoke an inherited project role from group on domain](#)
 - [Assign role to user on projects in a subtree](#)
 - [Assign role to group on projects in a subtree](#)
 - [Check if user has an inherited project role on project](#)
 - [Check if group has an inherited project role on project](#)
 - [Revoke an inherited project role from user on project](#)
 - [Revoke an inherited project role from group on project](#)
 - [List role assignments](#)
- [OS-PKI API](#)
 - [List revoked tokens](#)
- [Policies](#)
 - [Create policy](#)
 - [List policies](#)
 - [Show policy details](#)
 - [Update policy](#)
 - [Delete policy](#)
- [Projects](#)
 - [List projects](#)
 - [Create project](#)
 - [Show project details](#)
 - [Update project](#)
 - [Delete project](#)
- [Regions](#)
 - [Show region details](#)
 - [Update region](#)
 - [Delete region](#)
 - [List regions](#)
 - [Create region](#)
- [Roles](#)
 - [List roles](#)
 - [Create role](#)
 - [Show role details](#)
 - [Update role](#)
 - [Delete role](#)
 - [List role assignments for group on domain](#)
 - [Assign role to group on domain](#)
 - [Check whether group has role assignment on domain](#)

- [Unassign role from group on domain](#)
- [List role assignments for user on domain](#)
- [Assign role to user on domain](#)
- [Check whether user has role assignment on domain](#)
- [Unassigns role from user on domain](#)
- [List role assignments for group on project](#)
- [Assign role to group on project](#)
- [Check whether group has role assignment on project](#)
- [Unassign role from group on project](#)
- [List role assignments for user on project](#)
- [Assign role to user on project](#)
- [Check whether user has role assignment on project](#)
- [Unassign role from user on project](#)
- [List implied \(inference\) roles for role](#)
- [Create role inference rule](#)
- [Get role inference rule](#)
- [Confirm role inference rule](#)
- [Delete role inference rule](#)
- [List role assignments](#)
- [List all role inference rules](#)
- [Service catalog and endpoints](#)
 - [List services](#)
 - [Create service](#)
 - [Show service details](#)
 - [Update service](#)
 - [Delete service](#)
 - [List endpoints](#)
 - [Create endpoint](#)
 - [Show endpoint details](#)
 - [Update endpoint](#)
 - [Delete endpoint](#)
- [Users](#)
 - [List users](#)
 - [Create user](#)
 - [Show user details](#)
 - [Update user](#)
 - [Delete user](#)
 - [List groups to which a user belongs](#)
 - [List projects for user](#)
 - [Change password for user](#)

Identity API v2.0

Extensions

GET

/v2.0/extensions/{alias}

Show extension details

Shows details for an extension, by alias.

Normal response codes: 200,203

Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
alias	body	string	The alias for the extension. For example, “FOXNSOX”, “os- availability-zone”, “os-extended-quotas”, “os- share-unmanage” or “os-used-limits.”

Response Parameters

Name	In	Type	Description
x-openstack-request-id (Optional)	header	string	A unique request ID that provides tracking for the request. Provider must configure middleware to return a request ID header in a response.
alias	body	string	The alias for the extension. For example, “FOXNSOX”, “os- availability-zone”, “os- extended-quotas”, “os- share-unmanage” or “os- used-limits.”
updated	body	string	The date and time stamp when the extension was last updated.
description	body	string	The extension description.
name	body	string	The name of the extension. For example, “Fox In Socks.” example, “Fox In Socks.”

Response Example

```
{
  "extension": {
    "updated": "2013-07-07T12:00:0-00:00",
    "name": "OpenStack OAUTH1 API",
    "links": [
      {
        "href": "https://github.com/openstack/identity-api",
        "type": "text/html",
        "rel": "describedby"
      }
    ],
    "namespace": "https://docs.openstack.org/identity/api/ext/OS-OAUTH1/v1.0",
    "alias": "OS-OAUTH1",
    "description": "OpenStack OAuth 1.0a Delegated Auth Mechanism."
  }
}
```

GET

/v2.0/extensions

List extensions

Lists available extensions.

Normal response codes: 200,203

Error response codes: 413,405,404,403,401,400,503

Response Parameters

Name	In	Type	Description
x-openstack-request-id (Optional)	header	string	A unique request ID that provides tracking for the request. Provider must configure middleware to return a request ID header in

Name	In	Type	Description
			a response.
alias	body	string	The alias for the extension. For example, “FOXNSOX”, “os- availability-zone”, “os-extended-quotas”, “os- share-unmanage” or “os-used-limits.”
updated	body	string	The date and time stamp when the extension was last updated.
description	body	string	The extension description.
name	body	string	The name of the extension. For example, “Fox In Socks.”

Response Example

```
{
  "extensions": {
    "values": [
      {
        "updated": "2013-07-07T12:00:0-00:00",
        "name": "OpenStack S3 API",
        "links": [
          {
            "href": "https://github.com/openstack/identity-api",
            "type": "text/html",
            "rel": "describedby"
          }
        ],
        "namespace": "https://docs.openstack.org/identity/api/ext/s3tokens/v1.0",
        "alias": "s3tokens",
        "description": "OpenStack S3 API."
      },
      {
        "updated": "2013-07-23T12:00:0-00:00",
        "name": "OpenStack Keystone Endpoint Filter API",
        "links": [
          {
            "href": "https://github.com/openstack/identity-api/blob/master/openstack-identity-api/v3/src/markdown/identity-api-v3-os-ep-filter-ext.md",
            "type": "text/html",
            "rel": "describedby"
          }
        ],
        "namespace": "https://docs.openstack.org/identity/api/ext/OS-EP-FILTER/v1.0",
        "alias": "OS-EP-FILTER",
        "description": "OpenStack Keystone Endpoint Filter API."
      },
      {
        "updated": "2014-02-24T20:51:0-00:00",
        "name": "OpenStack Revoke API",
        "links": [
          {
            "href": "https://github.com/openstack/identity-api/blob/master/openstack-identity-api/v3/src/markdown/identity-api-v3-os-revoke-ext.md",
            "type": "text/html",
            "rel": "describedby"
          }
        ],
        "namespace": "https://docs.openstack.org/identity/api/ext/OS-REVOKE/v1.0",
        "alias": "OS-REVOKE",
        "description": "OpenStack revoked token reporting mechanism."
      },
      {
        "updated": "2013-12-17T12:00:0-00:00",
        "name": "OpenStack Federation APIs",
        "links": [
```

```

        {
            "href": "https://github.com/openstack/identity-api",
            "type": "text/html",
            "rel": "describedby"
        }
    ],
    "namespace": "https://docs.openstack.org/identity/api/ext/OS-FEDERATION/v1.0",
    "alias": "OS-FEDERATION",
    "description": "OpenStack Identity Providers Mechanism."
},
{
    "updated": "2013-07-11T17:14:00-00:00",
    "name": "OpenStack Keystone Admin",
    "links": [
        {
            "href": "https://github.com/openstack/identity-api",
            "type": "text/html",
            "rel": "describedby"
        }
    ],
    "namespace": "https://docs.openstack.org/identity/api/ext/OS-KSADM/v1.0",
    "alias": "OS-KSADM",
    "description": "OpenStack extensions to Keystone v2.0 API enabling
Administrative Operations."
},
{
    "updated": "2014-01-20T12:00:0-00:00",
    "name": "OpenStack Simple Certificate API",
    "links": [
        {
            "href": "https://github.com/openstack/identity-api",
            "type": "text/html",
            "rel": "describedby"
        }
    ],
    "namespace": "https://docs.openstack.org/identity/api/ext/OS-SIMPLE-
CERT/v1.0",
    "alias": "OS-SIMPLE-CERT",
    "description": "OpenStack simple certificate retrieval extension"
},
{
    "updated": "2013-07-07T12:00:0-00:00",
    "name": "OpenStack OAUTH1 API",
    "links": [
        {
            "href": "https://github.com/openstack/identity-api",
            "type": "text/html",
            "rel": "describedby"
        }
    ],
    "namespace": "https://docs.openstack.org/identity/api/ext/OS-OAUTH1/v1.0",
    "alias": "OS-OAUTH1",
    "description": "OpenStack OAuth 1.0a Delegated Auth Mechanism."
},
{
    "updated": "2013-07-07T12:00:0-00:00",
    "name": "OpenStack EC2 API",
    "links": [
        {
            "href": "https://github.com/openstack/identity-api",
            "type": "text/html",
            "rel": "describedby"
        }
    ],
    "namespace": "https://docs.openstack.org/identity/api/ext/OS-EC2/v1.0",
    "alias": "OS-EC2",
    "description": "OpenStack EC2 Credentials backend."
}
]
}

```

Tokens and tenants

GET

/v2.0/tenants

List tenants

Lists tenants to which the token has access.

Normal response codes: 200,

Error response codes: 413,405,404,403,401,400,503,

Response Parameters

Name	In	Type	Description
description	body	string	Description about the tenant.
tenants_links	body	array	Links of the tenants.
enabled	body	boolean	Indicates whether the tenant is enabled or disabled.
tenants	body	array	One or more tenant Objects.
id	body	string	The tenant ID.
name	body	string	The tenant name.

Response Example

```
{
  "tenants": [
    {
      "id": "1234",
      "name": "ACME Corp",
      "description": "A description ...",
      "enabled": true
    },
    {
      "id": "3456",
      "name": "Iron Works",
      "description": "A description ...",
      "enabled": true
    }
  ],
  "tenants_links": []
}
```

POST

/v2.0/tokens

Authenticate

Authenticates and generates a token.

The Identity API is a RESTful web service. It is the entry point to all service APIs. To access the Identity API, you must know its URL.

Each REST request against Identity requires the X-Auth-Token header. Clients obtain this token, along with the URL to other service APIs, by first authenticating against Identity with valid credentials.

To authenticate, you must provide either a user ID and password or a token.

If the authentication token has expired, this call returns the HTTP 401 status code.

If the token has expired, this call returns the HTTP 404 status code.

The Identity API treats expired tokens as no longer valid tokens.

The deployment determines how long expired tokens are stored.

To view the `trust` object, you need to set `trust` enable on the keystone configuration.

Normal response codes: 200,

Error response codes:413,405,404,403,401,400,503,

Request

Name	In	Type	Description
username (Optional)	body	string	The user name. Required if you include the <code>passwordCredentials</code> object. Otherwise, you must provide a token.
passwordCredentials (Optional)	body	string	A <code>passwordCredentials</code> object. To authenticate, you must provide either a user ID and password or a token.
tenantId (Optional)	body	string	The tenant ID. Both the <code>tenantId</code> and <code>tenantName</code> attributes are optional and mutually exclusive. If you specify both attributes, the server returns the <code>Bad Request (400)</code> response code.
token (Optional)	body	object	A <code>token</code> object. Required if you do not provide a password credential.
tenantName (Optional)	body	string	The tenant name. Both the <code>tenantId</code> and <code>tenantName</code> attributes are optional and mutually exclusive. If you specify both attributes, the server returns the <code>Bad Request (400)</code> response code.
password (Optional)	body	string	The password of the user. Required if you include the <code>passwordCredentials</code> object. Otherwise, you must provide a token.
id (Optional)	body	string	The token ID. This field is required in the <code>token</code> object.

Request Example

```
{
  "auth": {
    "tenantName": "demo",
    "token": {
      "id": "cbc36478b0bd8e67e89469c7749d4127"
    }
  }
}
```

Response Parameters

Name	In	Type	Description
impersonation (Optional)	body	boolean	The impersonation flag.
endpoints_links	body	array	Links for the endpoint.
serviceCatalog	body	array	List of <code>serviceCatalog</code> objects.
description	body	string	Description about the tenant.
type	body	string	Endpoint type.
expires	body	string	<p>The date and time when the token expires.</p> <p>The date and time stamp format is ISO 8601:</p> <p>CCYY-MM-DDThh:mm:ss.sssZ</p> <p>For example, 2015-08-27T09:49:58.000000Z.</p> <p>A <code>null</code> value indicates that the token never expires.</p>
enabled	body	boolean	Indicates whether the tenant is enabled or disabled.
name	body	string	The tenant name.
access	body	object	An <code>access</code> object.
trustee_user_id (Optional)	body	string	The trustee user ID.
token (Optional)	body	object	A <code>token</code> object. Required if you do not provide a password credential.
user	body	object	A <code>user</code> object, which shows the <code>username</code> , <code>roles_links</code> , <code>id</code> , <code>roles</code> , and <code>name</code> .
issued_at	body	string	<p>The date and time when the token was issued.</p> <p>The date and time stamp format is ISO 8601:</p> <p>CCYY-MM-DDThh:mm:ss.sssZ</p> <p>For example, 2015-08-27T09:49:58.000000Z.</p>
trustor_user_id (Optional)	body	string	The trustor user ID.
endpoints	body	array	One or more <code>endpoints</code> objects. Each object shows the <code>adminURL</code> , <code>region</code> , <code>internalURL</code> , <code>id</code> , and <code>publicURL</code> .

Name	In	Type	Description
			for the endpoint.
trust (Optional)	body	object	A trust object.
id	body	string	The tenant ID.
tenant	body	object	A tenant object.
metadata	body	object	A metadata object.

Response Example

```
{
  "access": {
    "token": {
      "issued_at": "2014-01-30T15:30:58.819584",
      "expires": "2014-01-31T15:30:58Z",
      "id": "aaaaa-bbbbb-cccc-dddd",
      "tenant": {
        "description": null,
        "enabled": true,
        "id": "fc394f2ab2df4114bde39905f800dc57",
        "name": "demo"
      }
    },
    "serviceCatalog": [
      {
        "endpoints": [
          {
            "adminURL":
"http://23.253.72.207:8774/v2/fc394f2ab2df4114bde39905f800dc57",
            "region": "RegionOne",
            "internalURL":
"http://23.253.72.207:8774/v2/fc394f2ab2df4114bde39905f800dc57",
            "id": "2dad48f09e2a447a9bf852bcd93548ef",
            "publicURL":
"http://23.253.72.207:8774/v2/fc394f2ab2df4114bde39905f800dc57"
          }
        ],
        "endpoints_links": [],
        "type": "compute",
        "name": "nova"
      },
      {
        "endpoints": [
          {
            "adminURL": "http://23.253.72.207:9696/",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:9696/",
            "id": "97c526db8d7a4c88bbb8d68db1bdcdb8",
            "publicURL": "http://23.253.72.207:9696/"
          }
        ],
        "endpoints_links": [],
        "type": "network",
        "name": "neutron"
      },
      {
        "endpoints": [
          {
            "adminURL":
"http://23.253.72.207:8776/v2/fc394f2ab2df4114bde39905f800dc57",
            "region": "RegionOne",
            "internalURL":
"http://23.253.72.207:8776/v2/fc394f2ab2df4114bde39905f800dc57",
            "id": "93f86dfcbbba143a39a33d0c2cd424870",
            "publicURL":
"http://23.253.72.207:8776/v2/fc394f2ab2df4114bde39905f800dc57"
          }
        ]
      }
    ]
  }
}
```

```

    ],
    "endpoints_links": [],
    "type": "volumev2",
    "name": "cinder"
  },
  {
    "endpoints": [
      {
        "adminURL": "http://23.253.72.207:8774/v3",
        "region": "RegionOne",
        "internalURL": "http://23.253.72.207:8774/v3",
        "id": "3eb274b12b1d47b2abc536038d87339e",
        "publicURL": "http://23.253.72.207:8774/v3"
      }
    ],
    "endpoints_links": [],
    "type": "compute3",
    "name": "nova"
  },
  {
    "endpoints": [
      {
        "adminURL": "http://23.253.72.207:3333",
        "region": "RegionOne",
        "internalURL": "http://23.253.72.207:3333",
        "id": "957f1e54afc64d33a62099faa5e980a2",
        "publicURL": "http://23.253.72.207:3333"
      }
    ],
    "endpoints_links": [],
    "type": "s3",
    "name": "s3"
  },
  {
    "endpoints": [
      {
        "adminURL": "http://23.253.72.207:9292",
        "region": "RegionOne",
        "internalURL": "http://23.253.72.207:9292",
        "id": "27d5749f36864c7d96bebf84a5ec9767",
        "publicURL": "http://23.253.72.207:9292"
      }
    ],
    "endpoints_links": [],
    "type": "image",
    "name": "glance"
  },
  {
    "endpoints": [
      {
        "adminURL":
"http://23.253.72.207:8776/v1/fc394f2ab2df4114bde39905f800dc57",
        "region": "RegionOne",
        "internalURL":
"http://23.253.72.207:8776/v1/fc394f2ab2df4114bde39905f800dc57",
        "id": "37c83a2157f944f1972e74658aa0b139",
        "publicURL":
"http://23.253.72.207:8776/v1/fc394f2ab2df4114bde39905f800dc57"
      }
    ],
    "endpoints_links": [],
    "type": "volume",
    "name": "cinder"
  },
  {
    "endpoints": [
      {
        "adminURL": "http://23.253.72.207:8773/services/Admin",
        "region": "RegionOne",
        "internalURL": "http://23.253.72.207:8773/services/Cloud",
        "id": "289b59289d6048e2912b327e5d3240ca",
        "publicURL": "http://23.253.72.207:8773/services/Cloud"
      }
    ],
    "endpoints_links": [],
    "type": "ec2",

```

```

        "name": "ec2"
    },
    {
        "endpoints": [
            {
                "adminURL": "http://23.253.72.207:8080",
                "region": "RegionOne",
                "internalURL":
"http://23.253.72.207:8080/v1/AUTH_fc394f2ab2df4114bde39905f800dc57",
                "id": "16b76b5e5b7d48039a6e4cc3129545f3",
                "publicURL":
"http://23.253.72.207:8080/v1/AUTH_fc394f2ab2df4114bde39905f800dc57"
            }
        ],
        "endpoints_links": [],
        "type": "object-store",
        "name": "swift"
    },
    {
        "endpoints": [
            {
                "adminURL": "http://example.com/identity_v2_admin",
                "region": "RegionOne",
                "internalURL": "http://example.com/identity",
                "id": "26af053673df4ef3a2340c4239e21ea2",
                "publicURL": "http://example.com/identity"
            }
        ],
        "endpoints_links": [],
        "type": "identity",
        "name": "keystone"
    }
],
"user": {
    "username": "demo",
    "roles_links": [],
    "id": "9a6590b2ab024747bc2167c4e064d00d",
    "roles": [
        {
            "name": "Member"
        },
        {
            "name": "anotherrole"
        }
    ],
    "name": "demo"
},
"metadata": {
    "is_admin": 0,
    "roles": [
        "7598ac3c634d4c3da4b9126a5f67ca2b",
        "f95c0ab82d6045d9805033ee1fbc80d4"
    ]
},
"trust": {
    "id": "394998fa61f14736b1f0c1f322882949",
    "trustee_user_id": "269348fdd9374b8885da1418e0730af1",
    "trustor_user_id": "3ec3164f750146be97f21559ee4d9c51",
    "impersonation": false
}
}

```

API versions

GET
/v2.0

Show version details

Shows details for the Identity API v2.0.

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Response Example

```
{
  "version": {
    "status": "stable",
    "updated": "2014-04-17T00:00:00Z",
    "media-types": [
      {
        "base": "application/json",
        "type": "application/vnd.openstack.identity-v2.0+json"
      }
    ],
    "id": "v2.0",
    "links": [
      {
        "href": "http://example.com/identity/v2.0/",
        "rel": "self"
      },
      {
        "href": "https://docs.openstack.org/",
        "rel": "describedby",
        "type": "text/html"
      }
    ]
  }
}
```

GET
/

List versions

Lists information about all Identity API versions.

Normal response codes: 200,300 Error response codes: 413,405,404,403,401,400,503

Response Example

```
{
  "versions": {
    "values": [
      {
        "id": "v3.4",
        "links": [
          {
            "href": "http://example.com/identity/v3/",
            "rel": "self"
          }
        ],
        "media-types": [
          {
            "base": "application/json",
            "type": "application/vnd.openstack.identity-v3+json"
          }
        ]
      }
    ]
  }
}
```

```

    },
    "status": "stable",
    "updated": "2015-03-30T00:00:00Z"
  },
  {
    "id": "v2.0",
    "links": [
      {
        "href": "http://example.com/identity/v2.0/",
        "rel": "self"
      },
      {
        "href": "https://docs.openstack.org/",
        "rel": "describedby",
        "type": "text/html"
      }
    ],
    "media-types": [
      {
        "base": "application/json",
        "type": "application/vnd.openstack.identity-v2.0+json"
      }
    ],
    "status": "stable",
    "updated": "2014-04-17T00:00:00Z"
  }
]
}

```

API Details

Overview

The OpenStack Identity API is implemented using a RESTful web service interface. All requests to authenticate and operate against the OpenStack Identity API should be performed using HTTPS.

OpenStack Identity enables clients to obtain tokens that permit access to OpenStack cloud services.

Intended audience

This reference is for software developers who develop applications that use the Identity API for authentication.

This reference assumes that the reader is familiar with RESTful web services, HTTP/1.1, and JSON serialization formats.

Identity concepts

To use OpenStack Identity, you must be familiar with these key concepts:

User

A digital representation of a person, system, or service that uses OpenStack cloud services. OpenStack Identity authentication services validate that an incoming request is being made by the user who claims to be making the call.

Users have a login and may be assigned tokens to access resources. Users may be directly assigned to a particular tenant and behave as if they are contained in that tenant.

Token

An arbitrary bit of text that is used to access resources. Each token has a scope that describes which resources are accessible with it. A token may be revoked at anytime and is valid for a finite duration.

While OpenStack Identity supports token-based authentication, the intention is for it to support additional protocols in the future. The intent is for it to be an integration service foremost, and not aspire to be a full-fledged identity store and management solution.

Credentials

Data that belongs to, is owned by, and generally only known by a user that the user can present to prove their identity.

Examples include:

- A matching username and password
- A matching username and API key
- A token that was issued to you

Authentication

In the context of the OpenStack Identity Service, the act of confirming the identity of a user or the truth of a claim. OpenStack Identity confirms that an incoming request is being made by the user who claims to be making the call by validating a set of identity information provided by the user.

These claims are initially in the form of a set of credentials (username & password, or username and API key). After initial confirmation, OpenStack Identity issues the user a token, which the user can then provide to demonstrate that their identity has been authenticated when making subsequent requests.

Tenant

A container used to group or isolate resources and/or identity objects. Depending on the service operator, a tenant can map to a customer, account, organization, or project.

Service

An OpenStack service, such as Compute (Nova), Object Storage (Swift), or Image Service (Glance). A service provides one or more endpoints through which users can access resources and perform operations.

Endpoint

A network-accessible address, usually described by a URL, where a service may be accessed. If using an extension for templates, you can create an endpoint template, which represents the templates of all the consumable services that are available across the regions.

Role

A personality that a user assumes when performing a specific set of operations. A role includes a set of rights and privileges. A user assuming that role inherits those rights and privileges.

In OpenStack Identity, a token that is issued to a user includes the list of roles that user can assume. Services that are being called by that user determine how they interpret the set of roles a user has and to which operations or resources each role grants access.

It is up to individual services such as the Compute service and Image service to assign meaning to these roles. As far as the Identity service is concerned, a role is an arbitrary name assigned by the user.

Paginated collections

To reduce load on the service, list operations return a maximum number of items at a time. The maximum number of items returned is determined by the Identity provider. To navigate the collection, you can set the `limit` and `marker` parameters in the URI. For example, `?limit=100&marker=1234`. The `marker` parameter is the ID of the last item in the previous list. Items are sorted by update time. When an update time is not available they are sorted by ID. The `limit` parameter sets the page size. Both parameters are optional. If the client requests a `limit` beyond that which is supported by the deployment a 413 response code may be thrown. A marker with an invalid ID returns a 404 response code.

Note

Paginated collections will never return a 404 error when the collection is empty - clients should expect an empty collection.

For convenience, collections contain atom `next` and `previous` links. The first page in the list does not contain a `previous` link, the last page in the list does not contain a `next` link. The following examples illustrate three pages in a collection of tenants. The first page was retrieved through a **GET** to `http://identity.api.openstack.org/v2.0/1234/tenants?limit=1`. In these examples, the `limit` parameter sets the page size to a single item. Subsequent `next` and `previous` links honor the initial page size. Thus, a client might follow links to traverse a paginated collection without having to input the `marker` parameter.

Example: Tenant collection, first page:

```
{
  "tenants": [
    {
      "id": "1234",
      "name": "ACME corp",
      "description": "A description ...",
      "enabled": true
    }
  ],
  "tenants_links": [
    {
      "rel": "next",
      "href": "http://identity.api.openstack.org/v2.0/tenants?limit=1&marker=1234"
    }
  ]
}
```

Example: Tenant collection, second page:

```
{
  "tenants": [
    {
      "id": "3645",
      "name": "Iron Works",
      "description": "A description ...",
      "enabled": true
    }
  ],
  "tenants_links": [
    {
      "rel": "next",
      "href": "http://identity.api.openstack.org/v2.0/tenants?limit=1&marker=3645"
    },
    {
      "rel": "previous",
      "href": "http://identity.api.openstack.org/v2.0/tenants?limit=1"
    }
  ]
}
```

Example: Tenant collection, last page:

```
{
  "tenants": [
    {
      "id": "9999",
      "name": "Bigz",
      "description": "A description ...",
      "enabled": true
    }
  ],
  "tenants_links": [
    {
      "rel": "previous",
      "href": "http://identity.api.openstack.org/v2.0/tenants?limit=1&marker=1234"
    }
  ]
}
```

Paginated collections contain a values property that contains the items in the collections. Links are accessed via the links property. The approach allows for extensibility of both the collection members and of the paginated collection itself. It also allows collections to be embedded in other objects as illustrated below. Here, a subset of groups are presented within a user. Clients must follow the `next` link to continue to retrieve additional groups belonging to a user.

Example: Paginated roles in user:

```
{
  "user": {
    "OS-ROLE:roles": [
      {
        "tenantId": "1234",
        "id": "Admin"
      },
      {
        "tenantId": "1234",
        "id": "DBUser"
      }
    ],
    "OS-ROLE:roles_links": [
      {
        "rel": "next",
        "href": "http://identity.api.openstack.org/v2.0/tenants/1234/users/u1000/roles?marker=Super"
      }
    ],
    "id": "u1000",
    "username": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

Request and response formats

The OpenStack Identity API only supports JSON data serialization request and response formats.

Use the `Content-Type` request header to specify the request format. This header is required for operations that have a request body.

The syntax for the `Content-Type` header is:

`Content-Type: application/json`

Use the `Accept` header to specify the response format:

Accept: application/json

If you do not specify a response format, the Accept header will be set to application/json by default.

Revocations

Allows the retrieval of revoked tokens.

GET

/v2.0/tokens/revoked

List Revoked Tokens (v2)

List the revoked tokens.

Normal response codes: 200

Error response codes: 400,401,403,404,405,413,503

Response Parameters

Name	In	Type	Description
signed	body	string	List of expired PKI tokens, signed by associated cryptographic message syntax (CMS).

Response Example

```
{
  "signed": "-----BEGIN
CMS-----\nMIICRAYJKoZIhvcNAQcCoIICNTCCAjECAQEExDTALBgIghkgBZQMEAgEwgZMGCSqG\nsIb3DQEHAaCBhQSBgn
sicmV2b2t1ZCI6IFt7ImV4cGlyZXMiOiAiMjAxNi0xMC0y\nnNVQyMjoxNjowMloiLCAiaWQiOiAiN2UyMTE1NDQxZTIzNG
JiYzhkYmU1ZGJmNTAz\nNjKxZWQiLCAiYXVkaXRfaWQiOiAicFlUY2h0OU9SdHl6VEo1ZHdmwWthZyJ9XX0x\nnggGFMIIB
gQIBATBcMFcxZzAJBgNVBAYTA1VTMQ4wDAYDVQQIDAVVbnNldDE0MAwG\nnA1UEBwwFVW5zZXQxDjAMBgNVBAoMBVuc2V0
MRgwFgYDVQQDDA93d3cuZXhhbXBs\nnZS5jb20CAQEwCwYJYIZIAWUDBAIBMA0GCSqGSIB3DQEBAQUABIIBAGv3q67FaZrF
\nnN6XUDtWio/uxjwKX5GVs1yqd37Wd/vLmjjsy0iZuslzKZnrkoffeLcu74I2sJh4x\nnFG8oQL99mck5Z9x8Uk8fLAI\nnW
a/pZnqN9m7oh8+JYNiRwZVrZq02fKEbH98JxB7\nnuessJSUyHie3AUMl5Zp/zhXg9Njf9s13kTh+XIg0eBMTdamqvtimTr
YMATIF+NF3\nn950E0FKQU5gzG7K+7JyJPT+/uzgC/tSckHTAxnB1YqI74FHckMSDCeJ5B7K+6yQ6\nn3Eq4ugsxY5UKaYg3
p/DXo4Muf7maMD+jhWL/mBqDAgPvmG50LSXLUB+/pyKXJULC\nnSszEkGHPzPI=\nn-----END CMS-----\n"
}
```

Identity Admin API v2.0

Tenants

GET

/v2.0/tenants/{tenantId}

Show tenant details, by ID

Shows details for a tenant, by ID.

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
tenantId	path	string	The tenant ID.

Response Parameters

Name	In	Type	Description
tenant	body	string	A tenant object.
enabled	body	boolean	Indicates whether the tenant is enabled or disabled.
description	body	string	The description of the tenant. If not set, this value is null.
name	body	string	Tenant name.
id	body	string	The tenant ID.

Response Example

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

GET

/v2.0/tenants/{tenantId}/users

List users on a tenant

Lists all users for a tenant.

Normal response codes: 200, 203 Error response codes: 413, 405, 404, 403, 401, 400, 503

Request

Name	In	Type	Description
tenantId	path	string	The tenant ID.

Response Parameters

Name	In	Type	Description
users	body	array	One or more <code>user</code> objects.
users_link	body	array	The link to the represented user collection.
enabled	body	boolean	Indicates whether the user is enabled (<code>true</code>) or disabled(<code>false</code>). The default value is <code>true</code> .
username	body	string	The user name.
email	body	string	The user email.
id	body	string	The user ID.

Response Example

```
{
  "users": [
    {
      "id": "3c9530e",
      "name": "admin",
      "email": "admin@example.org",
      "username": "admin",
      "enabled": true
    },
    {
      "id": "a0ae37b",
      "name": "demo",
      "email": "demo@example.org",
      "username": "demo",
      "enabled": true
    }
  ],
  "users_links": []
}
```

GET

/v2.0/tenants/{tenantId}/users/{userId}/roles

List roles for user

Lists roles for a user on a tenant. Excludes global roles.

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
userId	path	string	The user ID.
tenantId	path	string	The tenant ID.

Response Parameters

Name	In	Type	Description
roles	body	array	The collection of roles.
roles_links	body	array	The link to the represented role collection.
description	body	string	The role description.
name	body	string	The role name.
id	body	string	The role ID.

Response Example

```
{
  "roles": [
    {
      "id": "123",
      "name": "compute:admin",
      "description": "Nova Administrator"
    }
  ],
  "roles_links": []
}
```

GET

/v2.0/tenants

List tenants (admin endpoint)

Lists all tenants.

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Response Parameters

Name	In	Type	Description
tenants	body	array	The collection of tenants.
tenant_links	body	array	The link to the represented tenant collection.
enabled	body	boolean	Indicates whether the tenant is enabled or disabled.
description	body	string	The description of the tenant. If

Name	In	Type	Description
			not set, this value is null.
name	body	string	Tenant name.
id	body	string	The tenant ID.

Response Example

```
{
  "tenants": [
    {
      "id": "1234",
      "name": "ACME Corp",
      "description": "A description ...",
      "enabled": true
    },
    {
      "id": "3456",
      "name": "Iron Works",
      "description": "A description ...",
      "enabled": true
    }
  ],
  "tenants_links": []
}
```

GET

/v2.0/tenants

Show tenant details, by name

Shows details for a tenant, by name.

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
name (Optional)	query	string	Filters the response by a tenant name.

Response Parameters

Name	In	Type	Description
tenant	body	string	A tenant object.
enabled	body	boolean	Indicates whether the tenant is enabled or disabled.
description	body	string	The description of the tenant. If not set, this value is null.
name	body	string	Tenant name.
id	body	string	The tenant ID.

Response Example

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

DELETE

/v2.0/tenants/{tenantId}

Delete tenant

Deletes a tenant.

Normal response codes: 204 Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
tenantId	path	string	The tenant ID.

POST

/v2.0/tenants/{tenantId}

Update tenant

Updates a tenant.

Normal response codes: 200 Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
tenantId	path	string	The tenant ID.
tenant	body	string	A tenant object.
enabled	body	boolean	Indicates whether the tenant is enabled or disabled.
description	body	string	The description of the tenant. If not set, this value is null.
name	body	string	Tenant name.

Request Example

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

```
}
```

Response Parameters

Name	In	Type	Description
tenant	body	string	A tenant object.
enabled	body	boolean	Indicates whether the tenant is enabled or disabled.
description	body	string	The description of the tenant. If not set, this value is <code>null</code> .
name	body	string	Tenant name.
id	body	string	The tenant ID.

Response Example

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

POST

/v2.0/tenants

Create tenant

Creates a tenant.

Normal response codes: 201 Error response codes: 413,415,405,404,403,401,400,503,409

Request Example

```
{
  "tenant": {
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

Response Parameters

Name	In	Type	Description
tenant	body	string	A tenant object.
enabled	body	boolean	Indicates whether the tenant is enabled or disabled.
description	body	string	The description of the tenant. If not set, this value is <code>null</code> .

Name	In	Type	Description
name	body	string	Tenant name.
id	body	string	The tenant ID.

Tokens

GET

/v2.0/tokens/{tokenId}/endpoints

List endpoints for token

Lists the endpoints associated with a token.

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
tokenId	path	string	The authentication token for which to perform the operation.

Response Example

```
{
  "endpoints_links": [],
  "endpoints": [
    {
      "name": "nova",
      "adminURL": "https://nova.region-one.internal.com/v2/be1319401cfa4a0aa590b97cc7b64d8d",
      "region": "RegionOne",
      "internalURL": "https://nova.region-one.internal.com/v2/be1319401cfa4a0aa590b97cc7b64d8d",
      "type": "compute",
      "id": "11b41ee1b00841128b7333d4bf1a6140",
      "publicURL": "https://nova.region-one.public.com/v2/be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
      "name": "neutron",
      "adminURL": "https://neutron.region-one.internal.com/",
      "region": "RegionOne",
      "internalURL": "https://neutron.region-one.internal.com/",
      "type": "network",
      "id": "cdbfa3c416d741a9b5c968f2dc628acb",
      "publicURL": "https://neutron.region-one.public.com/"
    },
    {
      "name": "cinderv2",
      "adminURL": "https://cinderv2.region-one.internal.com/v2/be1319401cfa4a0aa590b97cc7b64d8d",
      "region": "RegionOne",
      "internalURL": "https://cinderv2.region-one.internal.com/v2/be1319401cfa4a0aa590b97cc7b64d8d",
      "type": "cinderv2v2",
      "id": "6de282e4132747ecb48f6fd8c525c6f6",
      "publicURL": "https://cinderv2.region-one.public.com/v2/be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
      "name": "trove",
```

```

        "adminURL": "https://trove.region-
one.internal.com/v1.0/be1319401cfa4a0aa590b97cc7b64d8d",
        "region": "RegionOne",
        "internalURL": "https://trove.region-
one.internal.com/v1.0/be1319401cfa4a0aa590b97cc7b64d8d",
        "type": "database",
        "id": "4bfad53a0c684bd981d093099eb7799b",
        "publicURL": "https://trove.region-
one.public.com/v1.0/be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
        "name": "s3",
        "adminURL": "https://s3.region-one.internal.com",
        "region": "RegionOne",
        "internalURL": "https://s3.region-one.internal.com",
        "type": "s3",
        "id": "50fb6b43cde44bb6b0e471a682dc42dd",
        "publicURL": "https://s3.region-one.public.com"
    },
    {
        "name": "glance",
        "adminURL": "https://glance.region-one.internal.com",
        "region": "RegionOne",
        "internalURL": "https://glance.region-one.internal.com",
        "type": "glance",
        "id": "838a338171164c3c8f56e6b5882267ff",
        "publicURL": "https://glance.region-one.public.com"
    },
    {
        "name": "novav3",
        "adminURL": "https://novav3.region-one.internal.com/v3",
        "region": "RegionOne",
        "internalURL": "https://novav3.region-one.internal.com/v3",
        "type": "compute",
        "id": "b437edd03d244bf4be605b9b8c8689e0",
        "publicURL": "https://novav3.region-one.public.com/v3"
    },
    {
        "name": "heat",
        "adminURL": "https://heat.region-one.internal.com/v1",
        "region": "RegionOne",
        "internalURL": "https://heat.region-one.internal.com/v1",
        "type": "cloudformation",
        "id": "7a0f6f37344d488fa596a1325e0fcf10",
        "publicURL": "https://heat.region-one.public.com/v1"
    },
    {
        "name": "cinder",
        "adminURL": "https://cinder.region-
one.internal.com/v1/be1319401cfa4a0aa590b97cc7b64d8d",
        "region": "RegionOne",
        "internalURL": "https://cinder.region-
one.internal.com/v1/be1319401cfa4a0aa590b97cc7b64d8d",
        "type": "cinderv2",
        "id": "d4f251065dce4ce3946d3c1b87e167f2",
        "publicURL": "https://cinder.region-
one.public.com/v1/be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
        "name": "ec2",
        "adminURL": "https://ec2.region-one.internal.com/services/Admin",
        "region": "RegionOne",
        "internalURL": "https://ec2.region-one.internal.com/services/Cloud",
        "type": "ec2",
        "id": "44c6bf28d9bd4d63bfb00d66f22439a8",
        "publicURL": "https://ec2.region-one.public.com/services/Cloud"
    },
    {
        "name": "heat",
        "adminURL": "https://heat.region-
one.internal.com/v1/be1319401cfa4a0aa590b97cc7b64d8d",
        "region": "RegionOne",
        "internalURL": "https://heat.region-
one.internal.com/v1/be1319401cfa4a0aa590b97cc7b64d8d",
        "type": "orchestration",
        "id": "21aaace3f33c46b8aaea2b17d41ffd54",

```

```

        "publicURL": "https://heat.region-
one.public.com/v1/be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
        "name": "swift",
        "adminURL": "https://swift.region-one.internal.com",
        "region": "RegionOne",
        "internalURL": "https://swift.region-
one.internal.com/v1/AUTH_be1319401cfa4a0aa590b97cc7b64d8d",
        "type": "object-store",
        "id": "35f7aca3be384580a3b1df43a97c2eb2",
        "publicURL": "https://swift.region-
one.public.com/v1/AUTH_be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
        "name": "keystone",
        "adminURL": "https://keystone.region-one.internal.com/v2.0",
        "region": "RegionOne",
        "internalURL": "https://keystone.region-one.internal.com/v2.0",
        "type": "identity",
        "id": "48da758fb58c47dcaf02000a4409a265",
        "publicURL": "https://keystone.region-one.public.com/v2.0"
    }
]
}

```

POST

/v2.0/tokens

Authenticate for admin API

Authenticates and generates a token.

A REST interface provides client authentication by using the POST method with `v2.0/tokens` as the path. Include a payload of credentials in the body.

The Identity API is a RESTful web service. It is the entry point to all service APIs. To access the Identity API, you must know its URL.

Each REST request against the Identity Service requires the `X-Auth-Token` header. Clients obtain this token and the URL endpoints for other service APIs by supplying their valid credentials to the authentication service.

If the authentication token has expired, this call returns the HTTP `unauthorized` (401) response code.

If the token has expired, this call returns the `itemNotFound` (404) response code.

The Identity API treats expired tokens as no longer valid tokens.

The deployment determines how long expired tokens are stored.

To view the `trust` object, you need to set `trust enable` on the keystone configuration.

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Request Example

```

{
  "auth": {
    "tenantName": "demo",
    "token": {
      "id": "cbc36478b0bd8e67e89469c7749d4127"
    }
  }
}

```

```
}  
  }  
}
```

Response Parameters

Name	In	Type	Description
token	body	string	A token object.
access	body	string	An access object.
issued_at	body	string	<p>The date and time when the token was issued.</p> <p>The date and time stamp format is ISO 8601:</p> <p>CCYY-MM-DDThh:mm:ss.sssZ</p> <p>For example, 2015-08-27T09:49:58.000000Z.</p>
expires	body	string	<p>The date and time when the token expires.</p> <p>The date and time stamp format is ISO 8601:</p> <p>CCYY-MM-DDThh:mm:ss.sssZ</p> <p>For example, 2015-08-27T09:49:58.000000Z.</p> <p>A null value indicates that the token never expires.</p>
id	body	string	The token ID.
tenant	body	string	A tenant object.
description	body	string	The description of the tenant. If not set, this value is null.
serviceCatalog	body	string	A serviceCatalog object.
type	body	string	The service catalog type.
endpoints_links	body	string	Links for the endpoint.
impersonation (Optional)	body	string	The impersonation flag.
endpoints	body	array	One or more endpoint objects. Each object shows the adminURL, region, internalURL, id, and publicURL for the endpoint.
user	body	string	A user object, which shows the username, roles_links, id, roles, and name.
roles	body	array	The collection of roles.
metadata	body	string	A metadata object.

Name	In	Type	Description
trust (Optional)	body	string	A trust object.

Response Example

```
{
  "access": {
    "token": {
      "issued_at": "2014-01-30T17:09:57.647795",
      "expires": "2014-01-31T17:09:57Z",
      "id": "admin_id",
      "tenant": {
        "description": null,
        "enabled": true,
        "id": "73f0aa26640f4971864919d0eb0f0880",
        "name": "admin"
      }
    },
    "serviceCatalog": [
      {
        "endpoints": [
          {
            "adminURL":
"http://23.253.72.207:8774/v2/73f0aa26640f4971864919d0eb0f0880",
            "region": "RegionOne",
            "internalURL":
"http://23.253.72.207:8774/v2/73f0aa26640f4971864919d0eb0f0880",
            "id": "2dad48f09e2a447a9bf852bcd93548ef",
            "publicURL":
"http://23.253.72.207:8774/v2/73f0aa26640f4971864919d0eb0f0880"
          }
        ],
        "endpoints_links": [],
        "type": "compute",
        "name": "nova"
      },
      {
        "endpoints": [
          {
            "adminURL": "http://23.253.72.207:9696/",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:9696/",
            "id": "97c526db8d7a4c88bbb8d68db1bdcdb8",
            "publicURL": "http://23.253.72.207:9696/"
          }
        ],
        "endpoints_links": [],
        "type": "network",
        "name": "neutron"
      },
      {
        "endpoints": [
          {
            "adminURL":
"http://23.253.72.207:8776/v2/73f0aa26640f4971864919d0eb0f0880",
            "region": "RegionOne",
            "internalURL":
"http://23.253.72.207:8776/v2/73f0aa26640f4971864919d0eb0f0880",
            "id": "93f86dfcbbba143a39a33d0c2cd424870",
            "publicURL":
"http://23.253.72.207:8776/v2/73f0aa26640f4971864919d0eb0f0880"
          }
        ],
        "endpoints_links": [],
        "type": "volumev2",
        "name": "cinder"
      },
      {
        "endpoints": [
          {

```

```

        "adminURL": "http://23.253.72.207:8774/v3",
        "region": "RegionOne",
        "internalURL": "http://23.253.72.207:8774/v3",
        "id": "3eb274b12b1d47b2abc536038d87339e",
        "publicURL": "http://23.253.72.207:8774/v3"
    },
    ],
    "endpoints_links": [],
    "type": "compute_v3",
    "name": "nova"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:3333",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:3333",
            "id": "957f1e54afc64d33a62099faa5e980a2",
            "publicURL": "http://23.253.72.207:3333"
        }
    ],
    "endpoints_links": [],
    "type": "s3",
    "name": "s3"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:9292",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:9292",
            "id": "27d5749f36864c7d96bebf84a5ec9767",
            "publicURL": "http://23.253.72.207:9292"
        }
    ],
    "endpoints_links": [],
    "type": "image",
    "name": "glance"
},
{
    "endpoints": [
        {
            "adminURL":
"http://23.253.72.207:8776/v1/73f0aa26640f4971864919d0eb0f0880",
            "region": "RegionOne",
            "internalURL":
"http://23.253.72.207:8776/v1/73f0aa26640f4971864919d0eb0f0880",
            "id": "37c83a2157f944f1972e74658aa0b139",
            "publicURL":
"http://23.253.72.207:8776/v1/73f0aa26640f4971864919d0eb0f0880"
        }
    ],
    "endpoints_links": [],
    "type": "volume",
    "name": "cinder"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8773/services/Admin",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8773/services/Cloud",
            "id": "289b59289d6048e2912b327e5d3240ca",
            "publicURL": "http://23.253.72.207:8773/services/Cloud"
        }
    ],
    "endpoints_links": [],
    "type": "ec2",
    "name": "ec2"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8080",
            "region": "RegionOne",
            "internalURL":

```

```

"http://23.253.72.207:8080/v1/AUTH_73f0aa26640f4971864919d0eb0f0880",
      "id": "16b76b5e5b7d48039a6e4cc3129545f3",
      "publicURL":
"http://23.253.72.207:8080/v1/AUTH_73f0aa26640f4971864919d0eb0f0880"
    }
  ],
  "endpoints_links": [],
  "type": "object-store",
  "name": "swift"
},
{
  "endpoints": [
    {
      "adminURL": "http://example.com/identity_v2_admin",
      "region": "RegionOne",
      "internalURL": "http://example.com/identity",
      "id": "26af053673df4ef3a2340c4239e21ea2",
      "publicURL": "http://example.com/identity"
    }
  ],
  "endpoints_links": [],
  "type": "identity",
  "name": "keystone"
}
],
"user": {
  "username": "admin",
  "roles_links": [],
  "id": "1f568815cb8148688e6ee9b2f7527dcc",
  "roles": [
    {
      "name": "service"
    },
    {
      "name": "admin"
    }
  ],
  "name": "admin"
},
"metadata": {
  "is_admin": 0,
  "roles": [
    "8341d3603a1d4d5985bfff09f10704d4d",
    "2e66d57df76946fdb034bc4da6fdec0"
  ]
},
"trust": {
  "id": "394998fa61f14736b1f0c1f322882949",
  "trustee_user_id": "269348fdd9374b8885da1418e0730af1",
  "trustor_user_id": "3ec3164f750146be97f21559ee4d9c51",
  "impersonation": false
}
}
}

```

GET

/v2.0/tokens/{tokenId}

Validate token

Validates a token and confirms that it belongs to a tenant.

Returns the permissions relevant to a particular client. Valid tokens are in the /tokens/{tokenId} path. If the token is not valid, this call returns the `itemNotFound` (404) response code. This method supports an optional parameter `belongsTo` to check the token scope against the ID of a project. If the token does not belong to the project specified in the parameter a `unauthorized` (401) response code will be returned.

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
tokenId	path	string	The authentication token for which to perform the operation.
belongsTo (Optional)	query	string	Project ID to check against token scope.

Response Example

```
{
  "access": {
    "token": {
      "id": "ab48a9efdfedb23ty3494",
      "expires": "2010-11-01T03:32:15-05:00",
      "tenant": {
        "id": "345",
        "name": "My Project"
      }
    },
    "user": {
      "id": "123",
      "name": "jqsmith",
      "roles": [
        {
          "id": "234",
          "name": "compute:admin"
        },
        {
          "id": "234",
          "name": "object-store:admin",
          "tenantId": "1"
        }
      ],
      "roles_links": []
    }
  }
}
```

HEAD

/v2.0/tokens/{tokenId}

Validate token (admin)

Validates a token and confirms that it belongs to a tenant, for performance. This method supports an optional parameter `belongsTo` to check the token scope against the ID of a project. If the token does not belong to the project specified in the parameter a `unauthorized (401)` response code will be returned.

Normal response codes: 200,203,204 Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
tokenId	path	string	The authentication token for which to perform the operation.
belongsTo (Optional)	query	string	Project ID to check against token scope.

DELETE

/v2.0/tokens/{tokenId}

Delete token

Deletes a token.

Normal response codes: 204 Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
tokenId	path	string	The authentication token for which to perform the operation.

Users

GET

/v2.0/users/{userId}/roles

List user global roles

Lists global roles for a user. Excludes tenant roles.

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
userId	path	string	The user ID.

Response Parameters

Name	In	Type	Description
roles	body	array	The collection of roles.
roles_links	body	array	The link to the represented role collection.
description	body	string	The role description.
name	body	string	The role name.
id	body	string	The role ID.

Response Example

```
{
  "roles": [
    {
      "id": "123",
      "name": "compute:admin",
      "description": "Nova Administrator"
    }
  ],
  "roles_links": []
}
```

POST

/v2.0/users

Create user (admin endpoint)

Creates a user.

Normal response codes: 201 Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
user	body	string	A user object, which shows the username, roles_links, id, roles, and name.
tenantId	path	string	The tenant ID.
password (Optional)	body	string	The user password.
enabled	body	boolean	Indicates whether the user is enabled (<code>true</code>) or disabled(<code>false</code>). The default value is <code>true</code> .
email	body	string	The user email.
name	body	string	The user name.
username (Optional)	body	string	The username of user.

Request Example

```
{
  "user": {
    "email": "new-user@example.com",
    "password": null,
    "enabled": true,
    "name": "new-user",
    "tenantId": "40429f980fac419bbfec372a5607c154"
  }
}
```

Response Parameters

Name	In	Type	Description
user	body	string	A user object, which shows the username, roles_links, id, roles, and name.
enabled	body	boolean	Indicates whether the user is enabled (<code>true</code>) or disabled(<code>false</code>). The default value is <code>true</code> .
email	body	string	The user email.
name	body	string	The user name.
username	body	string	The username of user.
id	body	string	The user ID.

GET

/v2.0/users

List users (admin endpoint)

Lists all users.

To show detailed information about a user by name, include the `name` query parameter in the request.

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Response Parameters

Name	In	Type	Description
users	body	array	One or more <code>user</code> objects.
enabled	body	boolean	Indicates whether the user is enabled (<code>true</code>) or disabled(<code>false</code>). The default value is <code>true</code> .
id	body	string	The user ID.
email	body	string	The user email.
name	body	string	The user name.
username	body	string	The username of user.

Response Example

```
{
  "users": [
    {
      "id": "3c9530e",
      "name": "admin",
      "email": "admin@example.org",
      "username": "admin",
      "enabled": true
    },
    {
      "id": "a0ae37b",
      "name": "demo",
      "email": "demo@example.org",
      "username": "demo",
      "enabled": true
    }
  ],
  "users_links": []
}
```

PUT

`/v2.0/users/{userId}`

Update user (admin endpoint)

Updates a user.

Normal response codes: 201 Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
userId	path	string	The user ID.
user	body	string	A user object, which shows the username, roles_links, id, roles, and name.
enabled	body	boolean	Indicates whether the user is enabled (true) or disabled(false). The default value is true.
email	body	string	The user email.
name	body	string	The user name.

Request Example

```
{
  "user": {
    "email": "updated_email@example.org",
    "tenantId": "1ca8e0"
  }
}
```

Response Parameters

Name	In	Type	Description
user	body	string	A user object, which shows the username, roles_links, id, roles, and name.
enabled	body	boolean	Indicates whether the user is enabled (true) or disabled(false). The default value is true.
email	body	string	The user email.
name	body	string	The user name.
username	body	string	The username of user.
id	body	string	The user ID.

Response Example

```
{
  "user": {
    "id": "a0ae37b",
    "name": "demo",
    "email": "updated_email@example.org",
    "username": "demo",
    "enabled": true,
    "tenantId": "1ca8e0"
  }
}
```

DELETE

/v2.0/users/{userId}

Delete user (admin endpoint)

Deletes a user.

Normal response codes: 204 Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
userId	path	string	The user ID.

GET

/v2.0/users/{userId}

Show user details (admin endpoint)

Shows details for a user, by ID.

The [openstack user show](#) command supports showing user details by name or ID. However, the command actually looks up the user ID for a user name and queries the user by ID.

As a workaround, complete these steps to show details for a user by name:

- [List all users](#).
- In the response, find the user name for which you want to show details and note its corresponding user ID.
- [Show details for user](#).

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
userId	body	string	The user ID.

Response Parameters

Name	In	Type	Description
user	body	string	A user object, which shows the username, roles_links, id, roles, and name.
enabled	body	boolean	Indicates whether the user is enabled (true) or disabled(false). The default value is true.
email	body	string	The user email.
name	body	string	The user name.

Name	In	Type	Description
username	body	string	The username of user.
id	body	string	The user ID.

Response Example

```
{
  "user": {
    "id": "3c9530e",
    "name": "admin",
    "email": "admin@example.org",
    "username": "admin",
    "enabled": true
  }
}
```

Endpoints

GET

/v2.0/endpoints

List endpoint templates

Normal response codes: 200

Response Parameters

Name	In	Type	Description
endpoints	body	array	One or more endpoint objects. Each object shows the adminURL, region, internalURL, id, and publicURL for the endpoint.

Response Example

```
{
  "endpoints": [
    {
      "adminurl": "http://host-1:8774/v1.1/$(tenant_id)s",
      "id": "8f9531231e044e218824b0e58688d262",
      "internalurl": "http://host-1:8774/v1.1/$(tenant_id)s",
      "publicurl": "http://host-1:8774/v1.1/$(tenant_id)s",
      "region": "RegionOne"
    },
    {
      "adminurl": "http://host-1:8774/v1.1/$(tenant_id)s",
      "id": "8f9531231e044e218824b0e58688d263",
      "internalurl": "http://host-1:8774/v1.1/$(tenant_id)s",
      "publicurl": "http://host-1:8774/v1.1/$(tenant_id)s",
      "region": "RegionOne"
    }
  ]
}
```

POST

/v2.0/endpoints

Create endpoint template

Normal response codes: 201

Request

Name	In	Type	Description
endpoint	body	object	Endpoint object. Contains publicURL, adminURL, internalURL, id and region for the endpoint.
region (Optional)	body	string	Region of the endpoint
service_id	body	string	Service ID
publicurl	body	string	Public URL
adminurl (Optional)	body	string	Admin URL
internalurl (Optional)	body	string	Internal URL

Request Example

```
{
  "endpoint": {
    "adminurl": null,
    "internalurl": null,
    "publicurl": "http://host-3:8774/v1.1/$(tenant_id)s",
    "region": "RegionOne",
    "service_id": "aea0aa3723e34ee3a5ac49ce86d4cc6e"
  }
}
```

Response Parameters

Name	In	Type	Description
endpoint	body	object	Endpoint object. Contains publicURL, adminURL, internalURL, id and region for the endpoint.
id	body	string	Endpoint ID
internalurl	body	string	Internal URL
publicurl	body	string	Public URL
region	body	string	Region of the endpoint

DELETE

/v2.0/endpoints/{endpointId}

Delete endpoint template

Normal response codes: 204

Request

Name	In	Type	Description
endpointId	path	string	Endpoint ID

Versions

GET

/v2.0

Get version details

Gets detailed information about a version of the Identity API.

Normal response codes: 200,203 Error response codes: 413,405,404,403,401,400,503

Response Parameters

Name	In	Type	Description
location	body	string	

Response Example

```
{
  "version": {
    "status": "stable",
    "updated": "2014-04-17T00:00:00Z",
    "media-types": [
      {
        "base": "application/json",
        "type": "application/vnd.openstack.identity-v2.0+json"
      }
    ],
    "id": "v2.0",
    "links": [
      {
        "href": "http://example.com/identity/v2.0/",
        "rel": "self"
      },
      {
        "href": "https://docs.openstack.org/",
        "rel": "describedby",
        "type": "text/html"
      }
    ]
  }
}
```

Certificates

Allows the retrieval of information for Certificate Authorities and certificates.

GET

/v2.0/certificates/ca

Show CA Certificate (v2)

Show the available CA certificate.

Normal response codes: 200

Error response codes: 401, 500

Response Example

```
MIIDgTCCAmmgAwIBAgIJAIr3n9+0RSC7MA0GCSqGSIb3DQEBCwUAMFcxZAJBgNV
BAYTA1VTMQ4wDAYDVQQIDAVVbnNldDEOMAwGA1UEBwwFVW5zZXQxZjAMBgNVBAom
BVVuc2V0MRgwFgYDVQQDDA93d3cuZXhhbXBsZS5jb20wHhcNMTYxMDIwMTMwMjE4
whcNMjYxMDE4MTMwMjE4WjBXMQswCQYDVQQGEwJVUzEOMAwGA1UECAwFVW5zZXQx
ZjAMBgNVBACMBVuc2V0MQ4wDAYDVQQKDAVVbnNldDEYMBYGA1UEAwwPd3d3LmV4
Yw1wbGUuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwoJkYpfJ
Bvqfq0eAuqTIZiunNQdnSUX/aMS5UuI6tjzSkYnR5FCdf9UP80rpA37gthvz3KK
XhNLqnnV8MLZEo3+1N5IAR+TE1foXnqGs6vNvj5Jn1lViXpIeaHxMwkJpJjPwxJ
nFLtXL1m9hIX5anV5ZyJwV8RIaMqnzOJ7QYiX07aouRvmtT501LQzr2ht214EzPY
YDt9UV/daSikrmroBnwgWmecafJOC1pxSyv02PANw+yhX6NHgGPJm0u0TSN2IK1p
o07ZVM3QJLLbEZFjcUK7FXNRk5ZfzjkCrJA110Ys3ByHTb2offffIyTYPuatQtff
0XvtIwMN5eIAswIDAQAB01AwTjAMBgNVHRMEBTADAQH/MB0GA1UdDgQWBbTZ4N1s
7DRmUBcrYhYDLSSDM0BCWzAFBgNVHSMEGDAwGwBTZ4N1s7DRmUBcrYhYDLSSDM0BC
WzANBgkqhkiG9w0BAQsFAAOCAQEALil6wvvi6yNVwu0zgt2iDYqHvnnHwnSVhEJ
eKeBFRxpuwiH+U0eygFB0/61D2r11cD0SdgaMfLAKKkSpQucJIsP3BYLWBj25oxn
NL2yB3HLZEeEbAQzXQwnRbwUbIpcp/XPlKjybiA3unqE+X/qdQZgxJ2Xgtp7bHhN
yzDCSOUZ1HrkKNXtFNvqRtoCeMBS2+jfqx2ap640RSnLihEi5710cUn2DbAR450I
+wppD5CcUTDsE0r+XbBK3Cm3dn6pVvVcawv5qDIdRB7JdsDbx6VC7gcBbdgdbLWz
Xf4KS8N77jeGjQKJ7QY5jkHdXhY+gGbeponch4y2VqLgMI0VGQ==
```

GET

/v2.0/certificates/signing

Show Signing Certificate (v2)

Show the available signing certificate.

Normal response codes: 200

Error response codes: 401, 500

Response Example

```
MIIDZjCCAk6gAwIBAgIBATANBgkqhkiG9w0BAQsFADEBXMQswCQYDVQQGEwJVUzE0
MAwGA1UECAwFVW5zZXQxZjAMBgNVBACMBVuc2V0MQ4wDAYDVQQKDAVVbnNldDEY
MBYGA1UEAwwPd3d3LmV4Yw1wbGUuY29tMB4XDTE2MTAyMDEzMDIwFjE2MTAx
ODEzMDIxOFowRZELMAKGA1UEBhMCMVVMxZjAMBgNVBAGMBVuc2V0MQ4wDAYDVQQK
DAVVbnNldDEYMBYGA1UEAwwPd3d3LmV4Yw1wbGUuY29tMIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEAua3cVYSD9KY31+wNXZv3HBS5MyzTfoY+nh4nJ2x8
Rame6liu4gkHYRonTuriIrgDLYo+2fuXrmyFcq1+8ke4KD3n24i8pzcrt6B0GAVYP
KdPyXU0EkZECNmH/tKjvVqMLHcq2apsZdZ5ujBtE5G4zbTjVIEzz90AbAmRVJy7S
seluCXBktg3IGa1WwqgU4B5pgog+VDpT8XPKFvHi1cVaX76qS6M0UxxA7ku0QUct
JxcyITS26Mxym7wOTI+7JV5A90w/dUN6CrGMrfHB59Pssx30s/BfoopFmIbbnHd00
ET0eifelkhWlWLFmmOHXwGYYX/aEyW3L/xCU5QDCz9B0wQIDAQAB000wSzAJBgNV
HRMEAIAAMB0GA1UdDgQWBbBQeoHszYSUSfGymk6kem/lpGVJS9DAfBgNVHSMEGDAW
gBTZ4N1s7DRmUBcrYhYDLSSDM0BCWzANBgkqhkiG9w0BAQsFAAOCAQEAfsh6AN7p
XWBg062LutpfDsRyXqOLYoFR4Y0Mzo1rH0jaozJsnOxsj42BdP+hBGjtZB9eUwgp
gx+MJQC4pz+Wuc/xMysDT6f0hyjZmsakXM92lsztLw7+Y7u9Ata2LDTER1Fv7X6D
I+kN+dhpqh01rIRWZvAf3TlZpEUG38cTxLD80sd0lq4BxSzmVkfQf4mcbu390X7i
0fGih0SxSa03idx9NWE0Ep9IaGLO/mfL84nb4YjgV9yJj+3CkxYvqPlpiM2rHD/C
hMgz/UB520xbjYjbWoyStZwvlSwKWY75C9iYA04TZrsh5UwvAT+I2Y2UY/krrZ2a
Rke2Bj7NAVXPHW==
```

Identity API v3

The Identity service generates authentication tokens that permit access to the OpenStack services REST APIs. Clients obtain this token and the URL endpoints for other service APIs by supplying their valid credentials to the authentication service.

Each time you make a REST API request to an OpenStack service, you supply your authentication token in the X-Auth-Token request header.

Like most OpenStack projects, OpenStack Identity protects its APIs by defining policy rules based on a role-based access control (RBAC) approach.

The Identity service configuration file sets the name and location of a JSON policy file that stores these rules.

For information about Identity API protection, see [Identity API protection with role-based access control \(RBAC\)](#) in the OpenStack Cloud Administrator Guide.

What's New in Version 3.8

- Allow a service user to fetch a token that has expired.
- Add a `password_expires_at` query parameter to user list and users in group list.

What's New in Version 3.7

- Addition of the `password_expires_at` field to the user response object.
- Introduce a flag to bypass expiration and revocation checking.

What's New in Version 3.6

- Listing role assignments for a tree of projects.
- Setting the project `is_domain` attribute enables a project to behave as a domain.
- Addition of the `is_domain` field to project scoped token response that represents whether a project is acting as a domain.
- Enable or disable a subtree in the project hierarchy.
- Delete a subtree in the project hierarchy.
- Additional identifier for tokens scoped to the designated `admin` project.
- Addition of `domain_id` filter to list user projects
- One role can imply another via `role_inference` rules.
- Enhance list role assignment to optionally provide names of entities.
- The defaults for domain-specific configuration options can be retrieved.
- Assignments can be specified as inherited, causing the assignment to be placed on any sub-projects.
- Support for domain specific roles.
- Support `enabled` and `id` as optional attributes to filter identity providers when listing.

What's New in Version 3.5

- Addition of `type` optional attribute to list credentials.
- Addition of `region_id` optional attribute to list endpoints.
- Addition of `is_domain` optional attribute to projects. Setting this currently has no effect, it is reserved for future use.

What's New in Version 3.4

- For tokenless authorization, the scope information may be set in the request headers.
- Addition of `parent_id` optional attribute to projects. This enables the construction of a hierarchy of projects.
- Addition of domain specific configuration management for a domain entity.
- Removal of `url` optional attribute for `regions`. This attribute was only used for the experimental phase of keystone-to-keystone federation and has been superseded by making service provider entries have its own entry in the service catalog.
- The JSON Home support now will indicate the status of resource if it is not stable and current.

What's New in Version 3.3

These features are considered stable as of September 4th, 2014.

- Addition of `name` optional variable to be included from service definition into the service catalog.
- Introduced a stand alone call to retrieve a service catalog.
- Introduced support for JSON Home.
- Introduced a standard call to retrieve possible project and domain scope targets for a token.
- Addition of `url` optional attribute for `regions`.

What's New in Version 3.2

These features are considered stable as of January 23, 2014.

- Introduced a mechanism to opt-out from catalog information during token validation
- Introduced a region resource for constructing a hierarchical container of groups of service endpoints
- Inexact filtering is supported on string attributes
- Listing collections may indicate only a subset of the data has been provided if a particular deployment has limited the number of entries a query may return

What's New in Version 3.1

These features are considered stable as of July 18, 2013.

- A token without an explicit scope of authorization is issued if the user does not specify a project and does not have authorization on the project specified by their default project attribute
- Introduced a generalized call for getting role assignments, with filtering for user, group, project, domain and role
- Introduced a mechanism to opt-out from catalog information during token creation
- Added optional bind information to token structure

What's New in Version 3.0

These features are considered stable as of February 20, 2013.

- Former “Service” and “Admin” APIs (including CRUD operations previously defined in the v2 OS-KSADM extension) are consolidated into a single core API
- “Tenants” are now known as “projects”
- “Groups”: a container representing a collection of users
- “Domains”: a high-level container for projects, users and groups
- “Policies”: a centralized repository for policy engine rule sets
- “Credentials”: generic credential storage per user (e.g. EC2, PKI, SSH, etc.)
- Roles can be granted at either the domain or project level
- User, group and project names only have to be unique within their owning domain
- Retrieving your list of projects (previously `GET /tenants`) is now explicitly based on your user ID: `GET /users/{user_id}/projects`
- Tokens explicitly represent user+project or user+domain pairs
- Partial updates are performed using the HTTP PATCH method
- Token ID values no longer appear in URLs

This page lists the Identity API operations in the following order:

- [Authentication and token management](#)
- [Credentials](#)
- [Domains](#)
- [Domain configuration](#)
- [Groups](#)
- [Policies](#)
- [Projects](#)
- [Regions](#)
- [Roles](#)
- [Service catalog and endpoints](#)
- [Users](#)
- [OS-INHERIT API](#)
- [OS-PKI API](#)

Authentication and token management

In exchange for a set of authentication credentials, the Identity service generates tokens. A token represents the authenticated identity of a user and, optionally, grants authorization on a specific project or domain.

The body of an authentication request must include a payload that specifies the authentication method, which is `password` or `token`, the credentials, and, optionally, the authorization scope. You can scope a token to a project or domain, or the token can be unscoped. You cannot scope a token to both a project and domain.

Tokens have IDs, which the Identity API returns in the `X-Subject-Token` response header.

Also, validates an authentication token and lists the domains, projects, roles, and endpoints to which the token gives access. Forces the immediate revocation of a token.

After you obtain an authentication token, you can:

- Make REST API requests to other OpenStack services. You supply the ID of your authentication token in the X-Auth-Token request header.
- Validate your authentication token and list the domains, projects, roles, and endpoints that your token gives you access to.
- Use your token to request another token scoped for a different domain and project.
- Force the immediate revocation of a token.
- List revoked public key infrastructure (PKI) tokens.

In v3.7 of the Identity API service, two new configuration options were added: `[resource] admin_project_name` and `[resource] admin_project_domain_name`. The options represent the project that only the cloud administrator should be able to access. When an authentication request for a token scoped to the admin project is processed, it will have an additional field in the token `{is_admin_project: True}`. The additional field can be used when writing policy rules that evaluate access control to APIs.

The Identity API treats expired tokens as no longer valid tokens. The deployment determines how long expired tokens are stored.

These authentication errors can occur:

Authentication errors

Response code	Description
Bad Request (400)	<p>The Identity service failed to parse the request as expected. One of the following errors occurred:</p> <ul style="list-style-type: none"> • A required attribute was missing. • An attribute that is not allowed was specified, such as an ID on a POST request in a basic CRUD operation. • An attribute of an unexpected data type was specified.
Unauthorized (401)	<p>One of the following errors occurred:</p> <ul style="list-style-type: none"> • Authentication was not performed. • The specified X-Auth-Token header is not valid. • The authentication credentials are not valid.
Forbidden (403)	The identity was successfully authenticated but it is not authorized to perform the requested action.
Not Found (404)	An operation failed because a referenced entity cannot be found by ID. For a POST request, the referenced entity might be specified in the request body rather than in the resource path.
Conflict (409)	<p>A POST or PATCH operation failed. For example, a client tried to update a unique attribute for an entity, which conflicts with that of another entity in the same collection.</p> <p>Or, a client issued a create operation twice on a collection with a user-defined, unique attribute. For example, a client made a POST <code>/users</code> request two times for the unique, user-defined name attribute for a user entity.</p>

POST

`/v3/auth/tokens`

Password authentication with unscoped authorization

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/auth_tokens

Authenticates an identity and generates a token. Uses the password authentication method. Authorization is unscoped.

The request body must include a payload that specifies the authentication method, which is `password`, and the user, by ID or name, and password credentials.

Normal response codes: 201

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
nocatalog (Optional)	query	string	(Since v3.1) The authentication response excludes the service catalog. By default, the response includes the service catalog.
domain	body	object	A domain object, containing:
name (Optional)	body	string	The user name. Required if you do not specify the ID of the user. If you specify the user name, you must also specify the domain, by ID or name.
auth	body	object	An auth object.
user	body	object	A user object.
password	body	object	The password object, contains the authentication information.
id (Optional)	body	string	The ID of the user. Required if you do not specify the user name.
identity	body	object	An identity object.
methods	body	array	The authentication method. For password authentication, specify <code>password</code> .

Request Example

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "admin",
          "domain": {
            "name": "Default"
          }
        }
      }
    }
  }
}
```

```

    }
  }
}

"password": "devstacker"

```

Response Parameters

Name	In	Type	Description
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.
domain	body	object	A domain object, containing:
methods	body	array	The authentication method. For password authentication, specify password.
expires_at	body	string	<p>The date and time when the token expires.</p> <p>The date and time stamp format is ISO 8601:</p> <p>CCYY-MM-DDThh:mm:ss.sssZ</p> <p>For example, 2015-08-27T09:49:58.000000Z.</p> <p>A null value indicates that the token never expires.</p>
token	body	object	A token object.
extras	body	object	A set of metadata key and value pairs, if any.
user	body	object	A user object.
audit_ids	body	array	A list of one or two audit IDs. An audit ID is a unique, randomly generated, URL-safe string that you can use to track a token. The first audit ID is the current audit ID for the token. The second audit ID is present for only re-scoped tokens and is the audit ID from the token before it was re-scoped. A re-scoped token is one that was exchanged for another token of the same or different scope. You can use these audit IDs to track the use of a token or chain of tokens across multiple requests and endpoints without exposing the token ID to non-privileged users.
issued_at	body	string	<p>The date and time when the token was issued.</p> <p>The date and time stamp format is ISO 8601:</p> <p>CCYY-MM-DDThh:mm:ss.sssZ</p> <p>For example, 2015-08-27T09:49:58.000000Z.</p>
id (Optional)	body	string	The ID of the user. Required if you do not specify the user name.
name (Optional)	body	string	The user name. Required if you do not specify the ID of the user. If you specify the user name, you must also specify the domain, by ID or name.

Response Example

```
{
  "token": {
    "methods": [
      "password"
    ],
    "expires_at": "2015-11-06T15:32:17.893769Z",
    "extras": {},
    "user": {
      "domain": {
        "id": "default",
        "name": "Default"
      },
      "id": "423f19a4ac1e4f48bbb4180756e6eb6c",
      "name": "admin",
      "password_expires_at": null
    },
    "audit_ids": [
      "ZzZwkUf1QfygX7pdYDBCQQ"
    ],
    "issued_at": "2015-11-06T14:32:17.893797Z"
  }
}
```

POST

/v3/auth/tokens

Password authentication with scoped authorization

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/auth_tokens

Authenticates an identity and generates a token. Uses the password authentication method and scopes authorization to a project or domain.

The request body must include a payload that specifies the `password` authentication method, the credentials, and the `project` or `domain` authorization scope.

Normal response codes: 201

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
nocatalog (Optional)	query	string	(Since v3.1) The authentication response excludes the service catalog. By default, the response includes the service catalog.
name (Optional)	body	string	The user name. Required if you do not specify the ID of the user. If you specify the user name, you must also specify the domain, by ID or name.
auth	body	object	An <code>auth</code> object.
user	body	object	A <code>user</code> object.
scope (Optional)	body	string	The authorization scope. (Since v3.4) Specify <code>unscoped</code> to make an explicit unscoped token request, which returns an unscoped response without any authorization. This request behaves the same as a token request with no scope where the user has no default project defined. If you do not

Name	In	Type	Description
			make an explicit unscoped token request and your role has a default project, the response might return a project- scoped token. If a default project is not defined, a token is issued without an explicit scope of authorization, which is the same as asking for an explicit unscoped token.
password	body	object	The password object, contains the authentication information.
id (Optional)	body	string	The ID of the user. Required if you do not specify the user name.
identity	body	object	An identity object.
methods	body	array	The authentication method. For password authentication, specify password .

Request Example

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "id": "ee4dfb6e5540447cb3741905149d9b6e",
          "password": "devstacker"
        }
      }
    },
    "scope": {
      "project": {
        "id": "a6944d763bf64ee6a275f1263fae0352"
      }
    }
  }
}
```

Response Parameters

Name	In	Type	Description
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.
domain	body	object	A domain object, containing:
region_id	body	string	(Since v3.2) The ID of the region that contains the service endpoint.
methods	body	array	The authentication method. For password authentication, specify password .
roles	body	array	A list of role objects, each containing:
url	body	string	The endpoint URL.
region	body	string	(Deprecated in v3.2) The geographic location of the service

Name	In	Type	Description
			endpoint.
token	body	object	A token object.
expires_at	body	string	<p>The date and time when the token expires.</p> <p>The date and time stamp format is ISO 8601:</p> <p>CCYY-MM-DDThh:mm:ss.sssZ</p> <p>For example, 2015-08-27T09:49:58.000000Z.</p> <p>A null value indicates that the token never expires.</p>
project	body	object	A project object, containing:
issued_at	body	string	<p>The date and time when the token was issued.</p> <p>The date and time stamp format is ISO 8601:</p> <p>CCYY-MM-DDThh:mm:ss.sssZ</p> <p>For example, 2015-08-27T09:49:58.000000Z.</p>
catalog	body	array	A catalog object.
extras	body	object	A set of metadata key and value pairs, if any.
user	body	object	A user object.
audit_ids	body	array	A list of one or two audit IDs. An audit ID is a unique, randomly generated, URL-safe string that you can use to track a token. The first audit ID is the current audit ID for the token. The second audit ID is present for only re-scoped tokens and is the audit ID from the token before it was re-scoped. A re-scoped token is one that was exchanged for another token of the same or different scope. You can use these audit IDs to track the use of a token or chain of tokens across multiple requests and endpoints without exposing the token ID to non-privileged users.
interface	body	string	The interface type, which describes the visibility of the endpoint. Value is: - <code>public</code> . Visible by end users on a publicly available network interface. - <code>internal</code> . Visible by end users on an unmetered internal network interface. - <code>admin</code> . Visible by administrative users on a secure network interface.
endpoints	body	array	A list of endpoint objects.
type	body	string	The endpoint type.
id (Optional)	body	string	The ID of the user. Required if you do not specify the user name.
name (Optional)	body	string	The user name. Required if you do not specify the ID of the user. If you specify the user name, you must also specify the domain, by ID or name.

POST

/v3/auth/tokens

Password authentication with explicit unscoped authorization

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/auth_tokens

Authenticates an identity and generates a token. Uses the password authentication method with explicit unscoped authorization.

The request body must include a payload that specifies the `password` authentication method, the credentials, and the `unscoped` authorization scope.

Normal response codes: 201

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
nocatalog (Optional)	query	string	(Since v3.1) The authentication response excludes the service catalog. By default, the response includes the service catalog.
name (Optional)	body	string	The user name. Required if you do not specify the ID of the user. If you specify the user name, you must also specify the domain, by ID or name.
auth	body	object	An <code>auth</code> object.
user	body	object	A <code>user</code> object.
scope (Optional)	body	string	The authorization scope. (Since v3.4) Specify <code>unscoped</code> to make an explicit unscoped token request, which returns an unscoped response without any authorization. This request behaves the same as a token request with no scope where the user has no default project defined. If you do not make an explicit <code>unscoped</code> token request and your role has a default project, the response might return a project- scoped token. If a default project is not defined, a token is issued without an explicit scope of authorization, which is the same as asking for an explicit unscoped token.
password	body	object	The <code>password</code> object, contains the authentication information.
id (Optional)	body	string	The ID of the user. Required if you do not specify the user name.
identity	body	object	An <code>identity</code> object.
methods	body	array	The authentication method. For password authentication, specify <code>password</code> .

Request Example

```
{
```

```

    "auth": {
      "identity": {
        "methods": [
          "password"
        ],
        "password": {
          "user": {
            "id": "ee4dfb6e5540447cb3741905149d9b6e",
            "password": "devstacker"
          }
        }
      },
      "scope": "unscoped"
    }
  }
}

```

Response Parameters

Name	In	Type	Description
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.
domain	body	object	A <code>domain</code> object, containing:
methods	body	array	The authentication method. For password authentication, specify <code>password</code> .
roles	body	array	A list of <code>role</code> objects, each containing:
expires_at	body	string	<p>The date and time when the token expires.</p> <p>The date and time stamp format is ISO 8601:</p> <p>CCYY-MM-DDThh:mm:ss.sssZ</p> <p>For example, 2015-08-27T09:49:58.000000Z.</p> <p>A <code>null</code> value indicates that the token never expires.</p>
token	body	object	A <code>token</code> object.
extras	body	object	A set of metadata key and value pairs, if any.
user	body	object	A <code>user</code> object.
audit_ids	body	array	A list of one or two audit IDs. An audit ID is a unique, randomly generated, URL-safe string that you can use to track a token. The first audit ID is the current audit ID for the token. The second audit ID is present for only re-scoped tokens and is the audit ID from the token before it was re-scoped. A re-scoped token is one that was exchanged for another token of the same or different scope. You can use these audit IDs to track the use of a token or chain of tokens across multiple requests and endpoints without exposing the token ID to non-privileged users.
issued_at	body	string	<p>The date and time when the token was issued.</p> <p>The date and time stamp format is ISO 8601:</p> <p>CCYY-MM-DDThh:mm:ss.sssZ</p> <p>For example, 2015-08-27T09:49:58.000000Z.</p>

Name	In	Type	Description
id (Optional)	body	string	The ID of the user. Required if you do not specify the user name.
name (Optional)	body	string	The user name. Required if you do not specify the ID of the user. If you specify the user name, you must also specify the domain, by ID or name.

POST

/v3/auth/tokens

Token authentication with unscoped authorization

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/auth_tokens

Authenticates an identity and generates a token. Uses the token authentication method. Authorization is unscoped.

In the request body, provide the token ID.

Normal response codes: 201

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
nocatalog (Optional)	query	string	(Since v3.1) The authentication response excludes the service catalog. By default, the response includes the service catalog.
identity	body	object	An <code>identity</code> object.
token	body	object	A <code>token</code> object. The token authentication method is used. This method is typically used in combination with a request to change authorization scope.
id	body	string	A token ID.
auth	body	object	An <code>auth</code> object.
methods	body	array	The authentication method. For token authentication, specify <code>token</code> .

Request Example

```
{
  "auth": {
    "identity": {
      "methods": [
        "token"
      ],
      "token": {
        "id": "'$OS_TOKEN'"
      }
    }
  }
}
```

```
}  
  }  
}
```

Response Parameters

Name	In	Type	Description
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.
X-Auth-Token	header	string	A valid authentication token for an administrative user.

POST

/v3/auth/tokens

Token authentication with scoped authorization

Relationship: https://docs.openstack.org/api/openstack-identity/3/re1/auth_tokens

Authenticates an identity and generates a token. Uses the token authentication method and scopes authorization to a project or domain.

In the request body, provide the token ID and the project or domain authorization scope.

Normal response codes: 201

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
nocatalog (Optional)	query	string	(Since v3.1) The authentication response excludes the service catalog. By default, the response includes the service catalog.
methods	body	array	The authentication method. For token authentication, specify token.
auth	body	object	An auth object.
token	body	object	A token object. The token authentication method is used. This method is typically used in combination with a request to change authorization scope.
audit_ids	body	array	A list of one or two audit IDs. An audit ID is a unique, randomly generated, URL-safe string that you can use to track a token. The first audit ID is the current audit ID for the token. The second audit ID is present for only re-scoped tokens and is the audit ID from the token before it was re-

Name	In	Type	Description
			scoped. A re- scoped token is one that was exchanged for another token of the same or different scope. You can use these audit IDs to track the use of a token or chain of tokens across multiple requests and endpoints without exposing the token ID to non-privileged users.
scope (Optional)	body	string	The authorization scope. (Since v3.4) Specify unscoped to make an explicit unscoped token request, which returns an unscoped response without any authorization. This request behaves the same as a token request with no scope where the user has no default project defined. If you do not make an explicit unscoped token request and your role has a default project, the response might return a project- scoped token. If a default project is not defined, a token is issued without an explicit scope of authorization, which is the same as asking for an explicit unscoped token.
id	body	string	A token ID.
identity	body	object	An identity object.

Request Example

```
{
  "auth": {
    "identity": {
      "methods": [
        "token"
      ],
      "token": {
        "id": "'$OS_TOKEN'"
      }
    },
    "scope": {
      "project": {
        "id": "5b50efd009b540559104ee3c03bbb2b7"
      }
    }
  }
}
```

Response Parameters

Name	In	Type	Description
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.

Name	In	Type	Description
X-Auth-Token	header	string	A valid authentication token for an administrative user.

GET

/v3/auth/tokens

Validate and show information for token

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/auth_tokens

Validates and shows information for a token, including its expiration date and authorization scope.

Pass your own token in the X-Auth-Token request header.

Pass the token that you want to validate in the X-Subject-Token request header.

Normal response codes: 200

Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
X-Auth-Token	header	string	A valid authentication token for an administrative user.
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.
nocatalog (Optional)	query	string	(Since v3.1) The authentication response excludes the service catalog. By default, the response includes the service catalog.
allow_expired (Optional)	query	bool	(Since v3.8) Allow fetching a token that has expired. By default expired tokens return a 404 exception.

Response Parameters

Name	In	Type	Description
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.
X-Auth-Token	header	string	A valid authentication token for an administrative user.
domain	body	object	A domain object, containing:

Name	In	Type	Description
methods	body	array	The authentication method, which is <code>password</code> , <code>token</code> , or both methods. Indicates the accumulated set of authentication methods that were used to obtain the token. For example, if the token was obtained by password authentication, it contains <code>password</code> . Later, if the token is exchanged by using the token authentication method one or more times, the subsequently created tokens contain both <code>password</code> and <code>token</code> in their <code>methods</code> attribute. Unlike multi-factor authentication, the <code>methods</code> attribute merely indicates the methods that were used to authenticate the user in exchange for a token. The client is responsible for determining the total number of authentication factors.
links	body	object	The links to the <code>domain</code> resource.
user	body	object	A <code>user</code> object.
token	body	object	A <code>token</code> object.
expires_at	body	string	The date and time when the token expires. The date and time stamp format is ISO 8601 : <code>CCYY-MM-DDThh:mm:ss.sssZ</code> For example, <code>2015-08-27T09:49:58.000000Z</code> . A <code>null</code> value indicates that the token never expires.
project	body	object	A <code>project</code> object, containing:
catalog	body	array	A <code>catalog</code> object.
extras	body	object	A set of metadata key and value pairs, if any.
roles	body	array	A list of <code>role</code> objects, each containing:
audit_ids	body	array	A list of one or two audit IDs. An audit ID is a unique, randomly generated, URL-safe string that you can use to track a token. The first audit ID is the current audit ID for the token. The second audit ID is present for only re-scoped tokens and is the audit ID from the token before it was re-scoped. A re-scoped token is one that was exchanged for another token of the same or different scope. You can use these audit IDs to track the use of a token or chain of tokens across multiple requests and endpoints without exposing the token ID to non-privileged users.
issued_at	body	string	The date and time when the token was issued. The date and time stamp format is ISO 8601 : <code>CCYY-MM-DDThh:mm:ss.sssZ</code> For example, <code>2015-08-27T09:49:58.000000Z</code> .

Name	In	Type	Description
id (Optional)	body	string	The ID of the user. Required if you do not specify the user name.
name (Optional)	body	string	The user name. Required if you do not specify the ID of the user. If you specify the user name, you must also specify the domain, by ID or name.

Response Example

```
{
  "token": {
    "methods": [
      "token"
    ],
    "expires_at": "2015-11-05T22:00:11.000000Z",
    "extras": {},
    "user": {
      "domain": {
        "id": "default",
        "name": "Default"
      },
      "id": "10a2e6e717a245d9acad3e5f97aeca3d",
      "name": "admin",
      "password_expires_at": null
    },
    "audit_ids": [
      "mAjXQhiYRyKwkB4qygdLVg"
    ],
    "issued_at": "2015-11-05T21:00:33.819948Z"
  }
}
```

HEAD

/v3/auth/tokens

Check token

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/auth_tokens

Validates a token.

This call is similar to GET /auth/tokens but no response body is provided even in the X-Subject-Token header.

The Identity API returns the same response as when the subject token was issued by POST /auth/tokens even if an error occurs because the token is not valid. An HTTP 204 response code indicates that the X-Subject-Token is valid.

Normal response codes: 200

Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
X-Auth-Token	header	string	A valid authentication token for an administrative user.

Name	In	Type	Description
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.
allow_expired (Optional)	query	bool	(Since v3.8) Allow fetching a token that has expired. By default expired tokens return a 404 exception.

DELETE

/v3/auth/tokens

Revoke token

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/auth_tokens

Revokes a token.

This call is similar to the HEAD /auth/tokens call except that the X-Subject-Token token is immediately not valid, regardless of the expires_at attribute value. An additional X-Auth-Token is not required.

Normal response codes: 204

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
X-Auth-Token	header	string	A valid authentication token for an administrative user.
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.

GET

/v3/auth/catalog

Get service catalog

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/auth_catalog

New in version 3.3

This call returns a service catalog for the X-Auth-Token provided in the request, even if the token does not contain a catalog itself (for example, if it was generated using ?nocatalog).

The structure of the catalog object is identical to that contained in a token.

Normal response codes: 204

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
X-Auth-Token	header	string	A valid authentication token for an administrative user.
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.

Response Parameters

Name	In	Type	Description
endpoints	body	array	A list of endpoint objects.
id	body	string	The UUID of the service to which the endpoint belongs.
type	body	string	The service type, which describes the API implemented by the service. Value is compute, ec2, identity, image, network, or volume.
name	body	string	The service name.

Response Example

```
{
  "catalog": [
    {
      "endpoints": [
        {
          "id": "39dc322ce86c4111b4f06c2eeae0841b",
          "interface": "public",
          "region": "RegionOne",
          "url": "http://localhost:5000"
        },
        {
          "id": "ec642f27474842e78bf059f6c48f4e99",
          "interface": "internal",
          "region": "RegionOne",
          "url": "http://localhost:5000"
        },
        {
          "id": "c609fc430175452290b62a4242e8a7e8",
          "interface": "admin",
          "region": "RegionOne",
          "url": "http://localhost:35357"
        }
      ],
      "id": "4363ae44bdf34a3981fde3b823cb9aa2",
      "type": "identity",
      "name": "keystone"
    }
  ],
  "links": {
    "self": "https://example.com/identity/v3/catalog",
    "previous": null,
    "next": null
  }
}
```

```
}
}
```

GET

/v3/auth/projects

Get available project scopes

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/auth_projects

New in version 3.3

This call returns the list of projects that are available to be scoped to based on the X-Auth-Token provided in the request.

The structure of the response is exactly the same as listing projects for a user.

Normal response codes: 204

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
X-Auth-Token	header	string	A valid authentication token for an administrative user.
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.

Response Parameters

Name	In	Type	Description
domain_id	body	string	The ID of the domain for the project.
enabled	body	boolean	If set to <code>true</code> , project is enabled. If set to <code>false</code> , project is disabled.
id	body	string	The ID for the project.
links	body	object	The links for the project resource.
name	body	string	The name of the project.

Response Example

```
{
  "projects": [
    {
```

```

    "domain_id": "1789d1",
    "enabled": true,
    "id": "263fd9",
    "links": {
      "self": "https://example.com/identity/v3/projects/263fd9"
    },
    "name": "Test Group"
  },
  {
    "domain_id": "1789d1",
    "enabled": true,
    "id": "50ef01",
    "links": {
      "self": "https://example.com/identity/v3/projects/50ef01"
    },
    "name": "Build Group"
  }
],
"links": {
  "self": "https://example.com/identity/v3/auth/projects",
  "previous": null,
  "next": null
}
}

```

GET

/v3/auth/domains

Get available domain scopes

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/auth_domains

New in version 3.3

This call returns the list of domains that are available to be scoped to based on the X-Auth-Token provided in the request.

The structure is the same as listing domains.

Normal response codes: 204

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
X-Auth-Token	header	string	A valid authentication token for an administrative user.
X-Subject-Token	header	string	The authentication token. An authentication response returns the token ID in this header rather than in the response body.

Response Parameters

Name	In	Type	Description
description	body	string	The description of the domain.
enabled	body	string	If set to <code>true</code> , domain is enabled. If set to <code>false</code> , domain is

Name	In	Type	Description
			disabled.
id	body	string	The ID of the domain.
links	body	object	The links to the domain resource.
name	body	string	The name of the domain.

Response Example

```
{
  "domains": [
    {
      "description": "my domain description",
      "enabled": true,
      "id": "1789d1",
      "links": {
        "self": "https://example.com/identity/v3/domains/1789d1"
      },
      "name": "my domain"
    },
    {
      "description": "description of my other domain",
      "enabled": true,
      "id": "43e8da",
      "links": {
        "self": "https://example.com/identity/v3/domains/43e8da"
      },
      "name": "another domain"
    }
  ],
  "links": {
    "self": "https://example.com/identity/v3/auth/domains",
    "previous": null,
    "next": null
  }
}
```

Credentials

In exchange for a set of authentication credentials that the user submits, the Identity service generates and returns a token. A token represents the authenticated identity of a user and, optionally, grants authorization on a specific project or domain.

You can list all credentials, and create, show details for, update, and delete a credential.

POST

/v3/credentials

Create credential

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/credentials>

Creates a credential.

The following example shows how to create an EC2-style credential. The credential blob is a string that

contains a JSON-serialized dictionary with the `access` and `secret` keys. This format is required when you specify the `ec2` type. To specify other credentials, such as `access_key`, change the type and contents of the data blob.

Normal response codes: 201

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
credential	body	object	A <code>credential</code> object.
project_id	body	string	The ID for the project.
type	body	string	The credential type, such as <code>ec2</code> or <code>cert</code> . The implementation determines the list of supported types.
blob	body	string	The credential itself, as a serialized blob.
user_id	body	string	The ID of the user who owns the credential.

Request Example

```
{
  "credential": {
    "blob": "{\"access\":\"181920\\\", \"secret\":\"secretKey\\\"}\",
    "project_id": "731fc6f265cd486d900f16e84c5cb594",
    "type": "ec2",
    "user_id": "bb5476fd12884539b41d5a88f838d773"
  }
}
```

Response Parameters

Name	In	Type	Description
credential	body	object	A <code>credential</code> object.
user_id	body	string	The ID of the user who owns the credential.
links	body	object	The links for the <code>credential</code> resource.
blob	body	string	The credential itself, as a serialized blob.
project_id	body	string	The ID for the project.
type	body	string	The credential type, such as <code>ec2</code> or <code>cert</code> . The implementation determines the list of supported types.
id	body	string	The UUID for the credential.

Response Example

```
{
  "credential": {
    "user_id": "bb5476fd12884539b41d5a88f838d773",
    "links": {
      "self":
"http://example.com/identity/v3/credentials/3d3367228f9c7665266604462ec60029bcd83ad89614021a80b2eb879c572510"
    },
    "blob": "{\"access\":\"181920\",\"secret\":\"secretKey\"}",
    "project_id": "731fc6f265cd486d900f16e84c5cb594",
    "type": "ec2",
    "id": "3d3367228f9c7665266604462ec60029bcd83ad89614021a80b2eb879c572510"
  }
}
```

GET

/v3/credentials

List credentials

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/credentials>

Lists all credentials.

Optionally, you can include the `user_id` query parameter in the URI to filter the response by a user.

Normal response codes: 200

Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
user_id (Optional)	query	string	Filters the response by a user ID.

Response Parameters

Name	In	Type	Description
user_id	body	string	The ID of the user who owns the credential.
links	body	object	The links for the <code>credentials</code> resource.
blob	body	string	The credential itself, as a serialized blob.
credentials	body	array	A list of <code>credential</code> objects.
project_id	body	string	The ID for the project.
type	body	string	The credential type, such as <code>ec2</code> or <code>cert</code> . The implementation determines the list of supported types.
id	body	string	The UUID for the credential.

Response Example

```
{
  "credentials": [
    {
      "user_id": "bb5476fd12884539b41d5a88f838d773",
      "links": {
        "self": "http://example.com/identity/v3/credentials/207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7"
      },
      "blob": "{\"access\": \"a42a27755ce6442596b049bd7dd8a563\", \"secret\": \"71faf1d40bb24c82b479b1c6fbdd9f0c\", \"trust_id\": null}",
      "project_id": "6e01855f345f4c59812999b5e459137d",
      "type": "ec2",
      "id": "207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7"
    },
    {
      "user_id": "6f556708d04b4ea6bc72d7df2296b71a",
      "links": {
        "self": "http://example.com/identity/v3/credentials/2441494e52ab6d594a34d74586075cb299489bdd1e9389e3ab06467a4f460609"
      },
      "blob": "{\"access\": \"7da79ff0aa364e1396f067e352b9b79a\", \"secret\": \"7a18d68ba8834b799d396f3ff6f1e98c\", \"trust_id\": null}",
      "project_id": "1a1d14690f3c4ec5bf5f321c5fde3c16",
      "type": "ec2",
      "id": "2441494e52ab6d594a34d74586075cb299489bdd1e9389e3ab06467a4f460609"
    },
    {
      "user_id": "c14107e65d5c4a7f8894fc4b3fc209ff",
      "links": {
        "self": "http://example.com/identity/v3/credentials/3397b204b5f04c495bc8c8f34c8a39996f280f9172658241873e15f070ec79d7"
      },
      "blob": "{\"access\": \"db9c58a558534a10a070110de4f9f20c\", \"secret\": \"973e790b88db447ba6f93bca02bc745b\", \"trust_id\": null}",
      "project_id": "7396e43183db40dcbf40dd727637b548",
      "type": "ec2",
      "id": "3397b204b5f04c495bc8c8f34c8a39996f280f9172658241873e15f070ec79d7"
    },
    {
      "user_id": "915cc5f8cca6466aba6c6be06cbabfd",
      "links": {
        "self": "http://example.com/identity/v3/credentials/352d5dd7a4aa19c4f2f23ee288bf65dc23a0bc293f40ffdd2128ffe6a8cf3e871"
      },
      "blob": "{\"access\": \"817c6c3487a440c1a0b1d3f92b30ca37\", \"secret\": \"47d681117d1c46e69a0c9ec811dae2e9\", \"trust_id\": null}",
      "project_id": "2bf9767f9db949ee8364262a28a23062",
      "type": "ec2",
      "id": "352d5dd7a4aa19c4f2f23ee288bf65dc23a0bc293f40ffdd2128ffe6a8cf3e871"
    },
    {
      "user_id": "bb5476fd12884539b41d5a88f838d773",
      "links": {
        "self": "http://example.com/identity/v3/credentials/3d3367228f9c7665266604462ec60029bcd83ad89614021a80b2eb879c572510"
      },
      "blob": "{\"access\": \"181920\", \"secret\": \"secretKey\"}",
      "project_id": "731fc6f265cd486d900f16e84c5cb594",
      "type": "ec2",
      "id": "3d3367228f9c7665266604462ec60029bcd83ad89614021a80b2eb879c572510"
    },
    {
      "user_id": "bb5476fd12884539b41d5a88f838d773",
      "links": {
        "self": "http://example.com/identity/v3/credentials/6b7d803fc03b85866904b6b79e0a8fa1f4013b584163b4477eed96717eb402c0"
      },
      "blob": "{\"access\": \"f2ba45670b504a518b46e920d760fde2\", \"secret\": \"bf7fff2b3a844730b2db793411756e55\", \"trust_id\": null}",
      "project_id": "731fc6f265cd486d900f16e84c5cb594",
      "type": "ec2",
      "id": "6b7d803fc03b85866904b6b79e0a8fa1f4013b584163b4477eed96717eb402c0"
    },
    {
      "user_id": "2b657f6742ac416697e6821b3b2ee785",
      "links": {
        "self": "http://example.com/identity/v3/credentials/7d391b869631e5c4836708ea3bb3e0a5cbe0481201b5f0ddd5685ad3b3faa564"
      },
      "blob": "{\"access\": \"a1525da4e7c0438ebf3058372d637b59\", \"secret\": \"c9165d2542b141e8b2a1ff61a5f5487c\", \"trust_id\": null}",
      "project_id": "2bf9767f9db949ee8364262a28a23062",
      "type": "ec2",
      "id": "7d391b869631e5c4836708ea3bb3e0a5cbe0481201b5f0ddd5685ad3b3faa564"
    },
    {
      "user_id": "bb5476fd12884539b41d5a88f838d773",
      "links": {
        "self": "http://example.com/identity/v3/credentials/7ef4faa904ae7b8b4ddc7bad15b05ee359dad7d7a9b82861d4ad92fdbbb2eb4e"
      },
      "blob": "{\"access\": \"7d7559359b57419eb5f5f5dcd65ab57d\", \"secret\": \"570652bcf8c2483c86eb29e9734eed3c\", \"trust_id\": null}",
      "project_id": "731fc6f265cd486d900f16e84c5cb594",
      "type": "ec2",
      "id": "7ef4faa904ae7b8b4ddc7bad15b05ee359dad7d7a9b82861d4ad92fdbbb2eb4e"
    },
    {
      "user_id": "aedb193e9bb8400485f8d8426f7a031f",
      "links": {
        "self": "http://example.com/identity/v3/credentials/9c1c428d8e0e8338a5e16489ecfff9962f2b00f984ce4c7e9015e4003f478df8"
      },
      "blob": "{\"access\": \"b3a6e5f4427c47e9b202264d91a19e49\", \"secret\": \"d9eb470f503f4b46932de38db7a79402\", \"trust_id\": null}",
      "project_id": "a2672ecf9dd34c6980448b25a47e0947",
      "type": "ec2",
      "id": "9c1c428d8e0e8338a5e16489ecfff9962f2b00f984ce4c7e9015e4003f478df8"
    },
    {
      "user_id": "c14107e65d5c4a7f8894fc4b3fc209ff",
      "links": {
        "self": "http://example.com/identity/v3/credentials/e2c35ac2becb0fca3c3c2f035692a4f46a9cbf3b6e86c8a47f5aafe837d78a05"
      },
      "blob": "{\"access\": \"1ed843b1bd4a409f9562400085adbaa4\", \"secret\": \"236ab24db1f04ec995fcf618ed4fc0f5\", \"trust_id\": null}",
      "project_id": "6e01855f345f4c59812999b5e459137d",
      "type": "ec2",
      "id": "e2c35ac2becb0fca3c3c2f035692a4f46a9cbf3b6e86c8a47f5aafe837d78a05"
    }
  ],
}
```

```
    "links": {
      "self": "http://example.com/identity/v3/credentials",
      "previous": null,
      "next": null
    }
  }
}
```

GET

/v3/credentials/{credential_id}

Show credential details

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/credential>

Shows details for a credential.

Normal response codes: 200

Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
credential_id	path	string	The UUID for the credential.

Response Parameters

Name	In	Type	Description
credential	body	object	A <code>credential</code> object.
user_id	body	string	The ID of the user who owns the credential.
links	body	object	The links for the <code>credential</code> resource.
blob	body	string	The credential itself, as a serialized blob.
project_id	body	string	The ID for the project.
type	body	string	The credential type, such as <code>ec2</code> or <code>cert</code> . The implementation determines the list of supported types.
id	body	string	The UUID for the credential.

Response Example

```
{
  "credential": {
    "user_id": "bb5476fd12884539b41d5a88f838d773",
    "links": {
      "self": "http://example.com/identity/v3/credentials/207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7"
    },
    "blob": "{\"access\": \"a42a27755ce6442596b049bd7dd8a563\", \"secret\": \"71faf1d40bb24c82b479b1c6fbdbd9f0c\", \"trust_id\": null}",
    "project_id": "6e01855f345f4c59812999b5e459137d",
    "type": "ec2",
    "id": "207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7"
  }
}
```

PATCH

/v3/credentials/{credential_id}

Update credential

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/credential>

Updates a credential.

Normal response codes: 200

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
credential_id	path	string	The UUID for the credential.
credential	body	object	A <code>credential</code> object.
project_id	body	string	The ID for the project.
type (Optional)	body	string	The credential type, such as <code>ec2</code> or <code>cert</code> . The implementation determines the list of supported types.
blob (Optional)	body	string	The credential itself, as a serialized blob.
user_id (Optional)	body	string	The ID of the user who owns the credential.

Request Example

```
{
  "credential": {
    "blob": "{\"access\":\"181920\",\"secret\":\"secretKey\"}",
    "project_id": "731fc6f265cd486d900f16e84c5cb594",
    "type": "ec2",
    "user_id": "bb5476fd12884539b41d5a88f838d773"
  }
}
```

Response Parameters

Name	In	Type	Description
credential	body	object	A <code>credential</code> object.
user_id	body	string	The ID of the user who owns the credential.
links	body	object	The links for the <code>credential</code> resource.
blob	body	string	The credential itself, as a serialized blob.

Name	In	Type	Description
project_id	body	string	The ID for the project.
type	body	string	The credential type, such as ec2 or cert. The implementation determines the list of supported types.
id	body	string	The UUID for the credential.

Response Example

```
{
  "credential": {
    "user_id": "bb5476fd12884539b41d5a88f838d773",
    "links": {
      "self":
"http://example.com/identity/v3/credentials/207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7"
    },
    "blob": "{\"access\":\"181920\\\", \"secret\\\": \"secretKey\\\"}\",
    "project_id": "731fc6f265cd486d900f16e84c5cb594",
    "type": "ec2",
    "id": "207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7"
  }
}
```

DELETE

/v3/credentials/{credential_id}

Delete credential

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/credential>

Deletes a credential.

Normal response codes: 204

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
credential_id	path	string	The UUID for the credential.

Domains

A domain is a collection of users, groups, and projects. Each group and project is owned by exactly one domain.

Each domain defines a namespace where certain API-visible name attributes exist, which affects whether those names must be globally unique or unique within that domain. In the Identity API, the uniqueness of these attributes is as follows:

- *Domain name.* Globally unique across all domains.
- *Role name.* Globally unique across all domains.
- *User name.* Unique within the owning domain.
- *Project name.* Unique within the owning domain.

- *Group name*. Unique within the owning domain.

GET

/v3/domains

List domains

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/domains>

Lists all domains.

Normal response codes: 200

Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
name (Optional)	query	string	Filters the response by a domain name.
enabled (Optional)	query	string	If set to true, then only domains that are enabled will be returned, if set to false only that are disabled will be returned. Any value other than 0, including no value, will be interpreted as true.

Response Parameters

Name	In	Type	Description
domains	body	array	A list of domain objects, each containing:
description	body	string	The description of the domain.
enabled	body	string	If set to true, domain is enabled. If set to false, domain is disabled.
id	body	string	The ID of the domain.
links	body	object	The links to the domain resource.
name	body	string	The name of the domain.

Response Example

```
{
  "domains": [
    {
      "description": "Used for swift functional testing",
      "enabled": true,
```

```

        "id": "5a75994a383c449184053ff7270c4e91",
        "links": {
            "self": "http://example.com/identity/v3/domains/5a75994a383c449184053ff7270c4e91"
        },
        "name": "swift_test"
    },
    {
        "description": "Owns users and tenants (i.e. projects) available on Identity API v2.",
        "enabled": true,
        "id": "default",
        "links": {
            "self": "http://example.com/identity/v3/domains/default"
        },
        "name": "Default"
    }
],
"links": {
    "next": null,
    "previous": null,
    "self": "http://example.com/identity/v3/domains"
}
}

```

POST

/v3/domains

Create domain

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/domains>

Creates a domain.

Normal response codes: 201

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
domain	body	object	A domain object, containing:
enabled (Optional)	body	string	<p>If set to <code>true</code>, domain is created enabled. If set to <code>false</code>, domain is created disabled. The default is <code>true</code>.</p> <p>Users can only authorize against an enabled domain (and any of its projects). In addition, users can only authenticate if the domain that owns them is also enabled. Disabling a domain prevents both of these things.</p>
description (Optional)	body	string	The description of the domain.
name	body	string	The name of the domain.

Request Example

```

{
    "domain": {
        "description": "Domain description",

```

```
    "enabled": true,  
    "name": "myDomain"  
  }  
}
```

Response Parameters

Name	In	Type	Description
domain	body	object	A domain object, containing:
description	body	string	The description of the domain.
enabled	body	string	If set to <code>true</code> , domain is enabled. If set to <code>false</code> , domain is disabled.
id	body	string	The ID of the domain.
links	body	object	The links to the domain resource.
name	body	string	The name of the domain.

GET

/v3/domains/{domain_id}

Show domain details

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/domains>

Shows details for a domain.

Normal response codes: 200

Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.

Response Parameters

Name	In	Type	Description
domain	body	object	A domain object, containing:
description	body	string	The description of the domain.
enabled	body	string	If set to <code>true</code> , domain is enabled. If set to <code>false</code> , domain is disabled.
id	body	string	The ID of the domain.
links	body	object	The links to the domain resource.

Name	In	Type	Description
name	body	string	The name of the domain.

Response Example

```
{
  "domain": {
    "description": "Owns users and tenants (i.e. projects) available on Identity API v2.",
    "enabled": true,
    "id": "default",
    "links": {
      "self": "http://example.com/identity/v3/domains/default"
    },
    "name": "Default"
  }
}
```

PATCH

/v3/domains/{domain_id}

Update domain

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/domain>

Updates a domain.

Normal response codes: 200

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
domain	body	object	A domain object, containing:
enabled (Optional)	body	string	<p>If set to <code>true</code>, domain is enabled. If set to <code>false</code>, domain is disabled. The default is <code>true</code>.</p> <p>Users can only authorize against an enabled domain (and any of its projects). In addition, users can only authenticate if the domain that owns them is also enabled. Disabling a domain prevents both of these things. When you disable a domain, all tokens that are authorized for that domain become no longer valid. If you reenable the domain, these tokens are not re-enabled.</p>
description (Optional)	body	string	The new description of the domain.
name (Optional)	body	string	The new name of the domain.

Request Example

```
{
  "domain": {
    "description": "Owns users and projects on Identity API v2."
  }
}
```

Response Parameters

Name	In	Type	Description
domain	body	object	A domain object, containing:
description	body	string	The description of the domain.
enabled	body	string	If set to <code>true</code> , domain is enabled. If set to <code>false</code> , domain is disabled.
id	body	string	The ID of the domain.
links	body	object	The links to the domain resource.
name	body	string	The name of the domain.

Response Example

```
{
  "domain": {
    "links": {
      "self": "http://example.com/identity/v3/domains/default"
    },
    "enabled": true,
    "description": "Owns users and projects on Identity API v2.",
    "name": "Default",
    "id": "default"
  }
}
```

DELETE

/v3/domains/{domain_id}

Delete domain

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/domain>

Deletes a domain.

To minimize the risk of accidentally deleting a domain, you must first disable the domain by using the update domain method.

When you delete a domain, this call also deletes all entities owned by it, such as users, groups, and projects, and any credentials and granted roles that relate to those entities.

If you try to delete an enabled domain, this call returns the **Forbidden (403)** response code.

Normal response codes: 204

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.

Domain configuration

You can manage domain-specific configuration options.

Domain-specific configuration options are structured within their group objects. The API supports only the `identity` and `ldap` groups. These groups override the default configuration settings for the storage of users and groups by the Identity server.

You can create, update, and delete domain-specific configuration options by using the HTTP PUT , PATCH , and DELETE methods. When updating, it is only necessary to include those options that are being updated.

To create an option, use the PUT method. The Identity API does not return options that are considered sensitive, although you can create and update these options. The only option currently considered sensitive is the `password` option within the `ldap` group.

The API enables you to include sensitive options as part of non- sensitive options. For example, you can include the `password` as part of the `url` option.

If you try to create or update configuration options for groups other than the `identity` or `ldap` groups, the **Forbidden (403)** response code is returned.

For information about how to integrate the Identity service with LDAP, see [Integrate Identity with LDAP](#).

GET

/v3/domains/config/default

Show default configuration settings

The default configuration settings for the options that can be overridden can be retrieved.

Relationship:: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config_default

Response Parameters

Name	In	Type	Description
config	body	object	A config object.
ldap	body	object	An ldap object. Required to set the LDAP group configuration options.
url	body	string	The LDAP URL.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from

Name	In	Type	Description
			where all users can be reached. For example, ou=Users, dc=root, dc=org.
identity	body	object	An identity object.
driver	body	string	The Identity backend driver.

Response Example

```
{
  "config": {
    "identity": {
      "driver": "ldap"
    },
    "ldap": {
      "url": "ldap://localhost",
      "user": "",
      "suffix": "cn=example,cn=com",
      ....
    }
  }
}
```

GET

/v3/domains/config/{group}/default

Show default configuration for a group

Reads the default configuration settings for a specific group.

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config_default

The API supports only the `identity` and `ldap` groups.

Normal response codes: 200

Error response codes: 413, 405, 404, 403, 401, 400, 503

Request

Name	In	Type	Description
group	path	string	The group ID.

Response Parameters

Name	In	Type	Description
ldap	body	object	An ldap object. Required to set the LDAP group configuration options.
url	body	string	The LDAP URL.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, ou=Users, dc=root, dc=org.

Name	In	Type	Description
identity	body	object	An identity object.
driver	body	string	The Identity backend driver.

Response Example

```
{
  "ldap": {
    "url": "ldap://localhost",
    "user": "",
    "suffix": "cn=example,cn=com".
    ....
  }
}
```

GET

/v3/domains/config/{group}/{option}/default

Show default option for a group

Reads the default configuration setting for an option within a group.

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config_default

The API supports only the `identity` and `ldap` groups. For the `ldap` group, a valid value is `url` or `user_tree_dn`. For the `identity` group, a valid value is `driver`.

Normal response codes: 200

Error response codes: 413, 405, 404, 403, 401, 400, 503

Request

Name	In	Type	Description
group	path	string	The group ID.
option	path	string	The option name. For the <code>ldap</code> group, a valid value is <code>url</code> or <code>user_tree_dn</code> . For the <code>identity</code> group, a valid value is <code>driver</code> .

Response Parameters

Name	In	Type	Description
url	body	string	The LDAP URL.
driver	body	string	The Identity backend driver.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, <code>ou=Users,dc=root,dc=org</code> .

Response Example

```
{
  "driver": "ldap"
}
```

GET

/v3/domains/{domain_id}/config/{group}/{option}

Show domain group option configuration

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config_default

Shows details for a domain group option configuration.

The API supports only the `identity` and `ldap` groups. For the `ldap` group, a valid value is `url` or `user_tree_dn`. For the `identity` group, a valid value is `driver`.

Normal response codes: 200

Error response codes: 413, 405, 404, 403, 401, 400, 503

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
group	path	string	The group ID.
option	path	string	The option name. For the <code>ldap</code> group, a valid value is <code>url</code> or <code>user_tree_dn</code> . For the <code>identity</code> group, a valid value is <code>driver</code> .

Response Parameters

Name	In	Type	Description
url	body	string	The LDAP URL.
driver	body	string	The Identity backend driver.
ldap	body	object	An <code>ldap</code> object. Required to set the LDAP group configuration options.
config	body	object	A <code>config</code> object.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, <code>ou=Users,dc=root,dc=org</code> .
identity	body	object	An <code>identity</code> object.

Response Example

```
{
  "url": "http://myldap/root"
}
```

PATCH

/v3/domains/{domain_id}/config/{group}/{option}

Update domain group option configuration

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config_default

Updates a domain group option configuration.

The API supports only the `identity` and `ldap` groups. For the `ldap` group, a valid value is `url` or `user_tree_dn`. For the `identity` group, a valid value is `driver`.

Normal response codes: 200

Error response codes: 413, 415, 405, 404, 403, 401, 400, 503, 409

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
group	path	string	The group ID.
option	path	string	The option name. For the <code>ldap</code> group, a valid value is <code>url</code> or <code>user_tree_dn</code> . For the <code>identity</code> group, a valid value is <code>driver</code> .
url	body	string	The LDAP URL.
driver	body	string	The Identity backend driver.
ldap	body	object	An <code>ldap</code> object. Required to set the LDAP group configuration options.
config	body	object	A <code>config</code> object.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, <code>ou=Users,dc=root,dc=org</code> .
identity	body	object	An <code>identity</code> object.

Request Example

```
{
  "url": "http://myldap/my_other_root"
}
```

Response Parameters

Name	In	Type	Description
url	body	string	The LDAP URL.
driver	body	string	The Identity backend driver.
ldap	body	object	An ldap object. Required to set the LDAP group configuration options.
config	body	object	A config object.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, ou=Users, dc=root, dc=org.
identity	body	object	An identity object.

Response Example

```
{
  "config": {
    "identity": {
      "driver": "keystone.identity.backends.ldap.Identity"
    },
    "ldap": {
      "url": "http://myldap/my_other_root",
      "user_tree_dn": "ou=Users, dc=my_new_root, dc=org"
    }
  }
}
```

DELETE

/v3/domains/{domain_id}/config/{group}/{option}

Delete domain group option configuration

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config_default

Deletes a domain group option configuration.

The API supports only the `identity` and `ldap` groups. For the `ldap` group, a valid value is `url` or `user_tree_dn`. For the `identity` group, a valid value is `driver`.

Normal response codes: 204

Error response codes: 413, 415, 405, 404, 403, 401, 400, 503, 409

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.

Name	In	Type	Description
group	path	string	The group ID.
option	path	string	The option name. For the ldap group, a valid value is url or user_tree_dn. For the identity group, a valid value is driver.

GET

/v3/domains/{domain_id}/config/{group}

Show domain group configuration

Relationship: https://docs.openstack.org/api/openstack-identity/3/re1/domain_config_default

Shows details for a domain group configuration.

The API supports only the identity and ldap groups.

Normal response codes: 200

Error response codes: 413, 405, 404, 403, 401, 400, 503

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
group	path	string	The group ID.

Response Parameters

Name	In	Type	Description
url	body	string	The LDAP URL.
driver	body	string	The Identity backend driver.
ldap	body	object	An ldap object. Required to set the LDAP group configuration options.
config	body	object	A config object.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, ou=Users, dc=root, dc=org.
identity	body	object	An identity object.

Response Example

```
{
```

```

    "ldap": {
      "url": "http://myldap/root",
      "user_tree_dn": "ou=Users,dc=root,dc=org"
    }
  }
}

```

PATCH

/v3/domains/{domain_id}/config/{group}

Update domain group configuration

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config_default

Updates a domain group configuration.

The API supports only the `identity` and `ldap` groups. If you try to set configuration options for other groups, this call fails with the `Forbidden (403)` response code.

Normal response codes: 200

Error response codes: 413, 415, 405, 404, 403, 401, 400, 503, 409

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
group	path	string	The group ID.
url	body	string	The LDAP URL.
driver	body	string	The Identity backend driver.
ldap	body	object	An ldap object. Required to set the LDAP group configuration options.
config	body	object	A config object.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, <code>ou=Users,dc=root,dc=org</code> .
identity	body	object	An identity object.

Request Example

```

{
  "config": {
    "ldap": {
      "url": "http://myldap/my_new_root",
      "user_tree_dn": "ou=Users,dc=my_new_root,dc=org"
    }
  }
}

```

Response Parameters

Name	In	Type	Description
url	body	string	The LDAP URL.
driver	body	string	The Identity backend driver.
ldap	body	object	An ldap object. Required to set the LDAP group configuration options.
config	body	object	A config object.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, ou=Users, dc=root, dc=org.
identity	body	object	An identity object.

Response Example

```
{
  "config": {
    "identity": {
      "driver": "keystone.identity.backends.ldap.Identity"
    },
    "ldap": {
      "url": "http://myldap/my_new_root",
      "user_tree_dn": "ou=Users, dc=my_new_root, dc=org"
    }
  }
}
```

DELETE

/v3/domains/{domain_id}/config/{group}

Delete domain group configuration

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config_default

Deletes a domain group configuration.

The API supports only the `identity` and `ldap` groups.

Normal response codes: 204

Error response codes: 413, 415, 405, 404, 403, 401, 400, 503, 409

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
group	path	string	The group ID.

Name	In	Type	Description

PUT

/v3/domains/{domain_id}/config

Create domain configuration

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config

Creates a domain configuration.

Normal response codes: 200, 201

Error response codes: 413, 405, 404, 403, 401, 400, 503

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
url	body	string	The LDAP URL.
driver	body	string	The Identity backend driver.
ldap	body	object	An ldap object. Required to set the LDAP group configuration options.
config	body	object	A config object.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, ou=Users,dc=root,dc=org.
identity	body	object	An identity object.

Request Example

```
{
  "config": {
    "identity": {
      "driver": "ldap"
    },
    "ldap": {
      "url": "ldap://myldap.com:389/",
      "user_tree_dn": "ou=Users,dc=my_new_root,dc=org"
    }
  }
}
```

Response Parameters

Name	In	Type	Description
url	body	string	The LDAP URL.
driver	body	string	The Identity backend driver.
ldap	body	object	An ldap object. Required to set the LDAP group configuration options.
config	body	object	A config object.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, ou=Users, dc=root, dc=org.
identity	body	object	An identity object.

Response Example

```
{
  "config": {
    "identity": {
      "driver": "ldap"
    },
    "ldap": {
      "url": "ldap://myldap.com:389/",
      "user_tree_dn": "ou=Users, dc=my_new_root, dc=org"
    }
  }
}
```

GET

/v3/domains/{domain_id}/config

Show domain configuration

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config

Shows details for a domain configuration.

Normal response codes: 200

Error response codes: 413, 405, 404, 403, 401, 400, 503

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.

Response Parameters

Name	In	Type	Description
url	body	string	The LDAP URL.

Name	In	Type	Description
driver	body	string	The Identity backend driver.
ldap	body	object	An ldap object. Required to set the LDAP group configuration options.
config	body	object	A config object.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, ou=Users,dc=root,dc=org.
identity	body	object	An identity object.

Response Example

```
{
  "config": {
    "identity": {
      "driver": "keystone.identity.backends.ldap.Identity"
    },
    "ldap": {
      "url": "http://myldap/root",
      "user_tree_dn": "ou=Users,dc=root,dc=org"
    }
  }
}
```

PATCH

/v3/domains/{domain_id}/config

Update domain configuration

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config

Updates a domain configuration.

Normal response codes: 200

Error response codes: 413, 415, 405, 404, 403, 401, 400, 503, 409

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
url	body	string	The LDAP URL.
driver	body	string	The Identity backend driver.
ldap	body	object	An ldap object. Required to set the LDAP group configuration options.
config	body	object	A config object.
user_tree_dn	body	string	The base distinguished name (DN) of

Name	In	Type	Description
			LDAP, from where all users can be reached. For example, ou=Users,dc=root,dc=org.
identity	body	object	An identity object.

Request Example

```
{
  "config": {
    "ldap": {
      "url": "http://myldap/my_new_root",
      "user_tree_dn": "ou=Users,dc=my_new_root,dc=org"
    }
  }
}
```

Response Parameters

Name	In	Type	Description
url	body	string	The LDAP URL.
driver	body	string	The Identity backend driver.
ldap	body	object	An ldap object. Required to set the LDAP group configuration options.
config	body	object	A config object.
user_tree_dn	body	string	The base distinguished name (DN) of LDAP, from where all users can be reached. For example, ou=Users,dc=root,dc=org.
identity	body	object	An identity object.

Response Example

```
{
  "config": {
    "identity": {
      "driver": "keystone.identity.backends.ldap.Identity"
    },
    "ldap": {
      "url": "http://myldap/my_new_root",
      "user_tree_dn": "ou=Users,dc=my_new_root,dc=org"
    }
  }
}
```

DELETE

/v3/domains/{domain_id}/config

Delete domain configuration

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_config

Deletes a domain configuration.

Normal response codes: 204

Error response codes: 413, 415, 405, 404, 403, 401, 400, 503, 409

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.

Groups

A group is a collection of users. Each group is owned by a domain.

You can use groups to ease the task of managing role assignments for users. Assigning a role to a group on a project or domain is equivalent to assigning the role to each group member on that project or domain.

When you unassign a role from a group, that role is automatically unassigned from any user that is a member of the group. Any tokens that authenticates those users to the relevant project or domain are revoked.

As with users, a group without any role assignments is useless from the perspective of an OpenStack service and has no access to resources. However, a group without role assignments is permitted as a way of acquiring or loading users and groups from external sources before mapping them to projects and domains.

GET
/v3/groups

List groups

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/groups>

Lists groups.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.

Code	Reason
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Request Parameters

Name	In	Type	Description
name (Optional)	query	string	Filters the response by a group name.
domain_id (Optional)	query	string	Filters the response by a domain ID.

Response Parameters

Name	In	Type	Description
links	body	object	The link to the collection of resources.
groups	body	array	A list of group objects, each containing:
description	body	string	The description of the group.
domain_id	body	string	The ID of the domain of the group.
id	body	string	The ID of the group.
links	body	object	The link to the resources in question.
name	body	string	The name of the group.

Response Example

```
{
  "links": {
    "self": "http://example.com/identity/v3/groups",
    "previous": null,
    "next": null
  },
  "groups": [
    {
      "description": "non-admin group",
      "domain_id": "default",
      "id": "96372bbb152f475aa37e9a76a25a029c",
      "links": {
        "self": "http://example.com/identity/v3/groups/96372bbb152f475aa37e9a76a25a029c"
      },
      "name": "nonadmins"
    },
    {
      "description": "openstack admin group",
      "domain_id": "default",
      "id": "9ce0ad4e58a84d7a97b92f7955d10c92",
      "links": {
        "self": "http://example.com/identity/v3/groups/9ce0ad4e58a84d7a97b92f7955d10c92"
      },
      "name": "admins"
    }
  ]
}
```

POST

/v3/groups

Create group

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/groups>

Creates a group.

Response Codes

Success

Code	Reason
201 - Created	Resource was created and is ready to use.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
group	body	object	A group object, containing:
description	body	string	The description of the group.
domain_id	body	string	The ID of the domain of the group.
name	body	string	The name of the group.

Request Example

```
{
  "group": {
    "description": "Contract developers",
    "domain_id": "default",
    "name": "Contract developers"
  }
}
```

Response Parameters

Name	In	Type	Description
group	body	object	A group object, containing:
description	body	string	The description of the group.
domain_id	body	string	The ID of the domain of the group.
id	body	string	The ID of the group.
links	body	object	The link to the resources in question.
name	body	string	The name of the group.

Response Example

```
{
  "group": {
    "description": "Contract developers",
    "domain_id": "default",
    "id": "c0d675eac29945ad9dfd08aa1bb75751",
    "links": {
      "self": "http://example.com/identity/v3/groups/c0d675eac29945ad9dfd08aa1bb75751"
    },
    "name": "Contract developers"
  }
}
```

GET

/v3/groups/{group_id}

Show group details

Relationship: <https://docs.openstack.org/api/openstack-identity/3/re1/group>

Shows details for a group.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Code	Reason
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
group_id	path	string	The group ID.

Response Parameters

Name	In	Type	Description
group	body	object	A group object, containing:
description	body	string	The description of the group.
domain_id	body	string	The ID of the domain of the group.
id	body	string	The ID of the group.
links	body	object	The link to the resources in question.
name	body	string	The name of the group.

Response Example

```
{
  "group": {
    "description": "Contract developers",
    "domain_id": "default",
    "id": "c0d675eac29945ad9dfd08aa1bb75751",
    "links": {
      "self": "http://example.com/identity/v3/groups/c0d675eac29945ad9dfd08aa1bb75751"
    },
    "name": "Contract developers"
  }
}
```

PATCH

/v3/groups/{group_id}

Update group

Relationship: <https://docs.openstack.org/api/openstack-identity/3/re1/group>

Updates a group.

If the back-end driver does not support this functionality, the call returns the Not Implemented (501) response code.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.
501 - Not Implemented	The server either does not recognize the request method, or it lacks the ability to fulfill the request.

Request Parameters

Name	In	Type	Description
group_id	path	string	The group ID.
group	body	object	A group object, containing:
description (Optional)	body	string	The new description of the group.
domain_id (Optional)	body	string	The ID of the new domain for the group. The ability to change the domain of a group is now deprecated, and will be removed in subsequent release. It is already disabled by default in most Identity service implementations.
name (Optional)	body	string	The new name of the group.

Request Example

```
{
  "group": {
    "description": "Contract developers 2016",
    "name": "Contract developers 2016"
  }
}
```

Response Parameters

Name	In	Type	Description
group	body	object	A group object, containing:
description	body	string	The description of the group.
domain_id	body	string	The ID of the domain of the group.
id	body	string	The ID of the group.
links	body	object	The link to the resources in question.
name	body	string	The name of the group.

Response Example

```
{
  "group": {
    "description": "Contract developers 2016",
    "domain_id": "default",
    "id": "c0d675eac29945ad9dfd08aa1bb75751",
    "links": {
      "self": "http://example.com/identity/v3/groups/c0d675eac29945ad9dfd08aa1bb75751"
    },
    "name": "Contract developers 2016"
  }
}
```

DELETE

/v3/groups/{group_id}

Delete group

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/group>

Deletes a group.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before

Code	Reason
	making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
group_id	path	string	The group ID.

GET

/v3/groups/{group_id}/users

List users in group

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/group_users

Lists the users that belong to a group.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
group_id	path	string	The group ID.

Name	In	Type	Description
password_expires_at (Optional)	query	string	<p>Filter results based on which user passwords have expired. The query should include an operator and a timestamp with a colon (:) separating the two, for example:</p> <p>password_expires_at={operator}:{timestamp}</p> <ul style="list-style-type: none"> Valid operators are: lt, lte, gt, gte, eq, and neq <ul style="list-style-type: none"> lt: expiration time lower than the timestamp lte: expiration time lower than or equal to the timestamp gt: expiration time higher than the timestamp gte: expiration time higher than or equal to the timestamp eq: expiration time equal to the timestamp neq: expiration time not equal to the timestamp Valid timestamps are of the form: YYYY-MM-DDTHH:mm:ssZ. <p>For example:</p> <p>/v3/users?password_expires_at=lt:2016-12-08T22:02:00Z</p> <p>The example would return a list of users whose password expired before the timestamp (2016-12-08T22:02:00Z).</p>

Response Example

```
{
  "links": {
    "self":
"http://example.com/identity/v3/groups/9ce0ad4e58a84d7a97b92f7955d10c92/users",
    "previous": null,
    "next": null
  },
  "users": [
    {
      "domain_id": "default",
      "description": null,
      "enabled": true,
      "id": "acd565a08293c1e48bc0dd0d72ad5d5d"
      "name": "Henry",
      "links": {
        "self":
"http://example.com/identity/v3/users/acd565a08293c1e48bc0dd0d72ad5d5d"
      }
    },
    {
      "domain_id": "default",
      "description": null,
      "enabled": true,
      "id": "fff603a0829d41e48bc0dd0d72ad61ce",
      "name": "Paul",
      "links": {
        "self":
"http://example.com/identity/v3/users/fff603a0829d41e48bc0dd0d72ad61ce"
      },
      "password_expires_at": "2016-11-06T15:32:17.000000"
    }
  ]
}
```

PUT

/v3/groups/{group_id}/users/{user_id}

Add user to group

Relationship: <https://docs.openstack.org/api/openstack->

identity/3/rel/group_user

Adds a user to a group.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
user_id	path	string	The user ID.
group_id	path	string	The group ID.

HEAD

/v3/groups/{group_id}/users/{user_id}

Check whether user belongs to group

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/group_user

Validates that a user belongs to a group.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
user_id	path	string	The user ID.
group_id	path	string	The group ID.

DELETE

/v3/groups/{group_id}/users/{user_id}

Remove user from group

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/group_user

Removes a user from a group.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user

Code	Reason
	to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
user_id	path	string	The user ID.
group_id	path	string	The group ID.

OS-INHERIT API

Enables projects to inherit role assignments from either their owning domain or projects that are higher in the hierarchy.

(Since API v3.4) The OS-INHERIT extension allows inheritance from both projects and domains. To access project inheritance, the Identity service server must run at least API v3.4.

PUT

/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects

Assign role to user on projects owned by domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/domain_user_role_inherited_to_projects

Assigns a role to a user in projects owned by a domain.

The inherited role is only applied to the owned projects (both existing and future projects), and will not appear as a role in a domain scoped token.

Normal response codes: 204

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
role_id	path	string	The role ID.
user_id	path	string	The user ID.

PUT

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

Assign role to group on projects owned by a domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/domain_group_role_inherited_to_projects

The inherited role is only applied to the owned projects (both existing and future projects), and will not appear as a role in a domain scoped token.

Normal response codes: 204

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
group_id	path	string	The role ID.
role_id	path	string	The user ID.

GET

/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/inherited_to_projects

List user's inherited project roles on a domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/domain_user_roles_inherited_to_projects

The list only contains those role assignments to the domain that were specified as being inherited to projects within that domain.

Normal response codes: 200

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
user_id	path	string	The user ID.

Response Example

```
{
  "roles": [
    {
      "id": "91011",
      "links": {
        "self": "http://example.com/identity/v3/roles/91011"
      },
      "name": "admin"
    },
    {
      "id": "91011",
      "links": {
        "self": "http://example.com/identity/v3/roles/91011"
      },
      "name": "admin"
    }
  ],
  "links": {
    "self": "http://example.com/identity/v3/OS-INHERIT/domains/1234/users/5678/roles/inherited_to_projects",
    "previous": null,
    "next": null
  }
}
```

```
}  
}
```

GET

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

List group's inherited project roles on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/domain_group_roles_inherited_to_projects

The list only contains those role assignments to the domain that were specified as being inherited to projects within that domain.

Normal response codes: 200

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
group_id	path	string	The group ID.

Response Example

```
{  
  "roles": [  
    {  
      "id": "91011",  
      "links": {  
        "self": "http://example.com/identity/v3/roles/91011"  
      },  
      "name": "admin"  
    },  
    {  
      "id": "91011",  
      "links": {  
        "self": "http://example.com/identity/v3/roles/91011"  
      },  
      "name": "admin"  
    }  
  ],  
  "links": {  
    "self": "http://example.com/identity/v3/OS-INHERIT/domains/1234/groups/5678/roles/inherited_to_projects",  
    "previous": null,  
    "next": null  
  }  
}
```

HEAD

/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects

Check if user has an inherited project role on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/domain_user_role_inherited_to_projects

Checks whether a user has an inherited project role in a domain.

Normal response codes: 204

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
role_id	path	string	The role ID.
user_id	path	string	The user ID.

HEAD

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

Check if group has an inherited project role on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/domain_group_role_inherited_to_projects

Checks whether a group has an inherited project role in a domain.

Normal response codes: 204

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
group_id	path	string	The group ID.
role_id	path	string	The role ID.

DELETE

/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects

Revoke an inherited project role from user on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/domain_user_role_inherited_to_projects

Revokes an inherited project role from a user in a domain.

Normal response codes: 204

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
role_id	path	string	The role ID.
user_id	path	string	The user ID.

DELETE

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

Revoke an inherited project role from group on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/domain_group_role_inherited_to_projects

Revokes an inherited project role from a group in a domain.

Normal response codes: 204

Request

Name	In	Type	Description
domain_id	path	string	The domain ID.
group_id	path	string	The group ID.
role_id	path	string	The role ID.

PUT

/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects

Assign role to user on projects in a subtree

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/project_user_role_inherited_to_projects

The inherited role assignment is anchored to a project and applied to its subtree in the projects hierarchy (both existing and future projects).

- Note: It is possible for a user to have both a regular (non-inherited) and an inherited role assignment on the same project.
- Note: The request doesn't require a body, which will be ignored if provided.

Normal response codes: 204

Request

Name	In	Type	Description
project_id	body	string	The ID for the project.
role_id	path	string	The role ID.
user_id	path	string	The user ID.

PUT

/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects

Assign role to group on projects in a subtree

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/project_group_role_inherited_to_projects

The inherited role assignment is anchored to a project and applied to its subtree in the projects hierarchy (both existing and future projects).

- Note: It is possible for a group to have both a regular (non-inherited) and an inherited role assignment on the same project.
- Note: The request doesn't require a body, which will be ignored if provided.

Normal response codes: 204

Request

Name	In	Type	Description
group_id	path	string	The group ID.
project_id	path	string	The project ID.
role_id	path	string	The role ID.

HEAD

/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects

Check if user has an inherited project role on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/project_user_role_inherited_to_projects

Checks whether a user has a role assignment with the `inherited_to_projects` flag in a project.

Normal response codes: 200

Request

Name	In	Type	Description
project_id	path	string	The project ID.
role_id	path	string	The role ID.
user_id	path	string	The user ID.

HEAD

/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects

Check if group has an inherited project role on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/project_group_role_inherited_to_projects

Checks whether a group has a role assignment with the `inherited_to_projects` flag in a project.

Normal response codes: 200

Request

Name	In	Type	Description
group_id	path	string	The group ID.
project_id	path	string	The project ID.
role_id	path	string	The role ID.

DELETE

/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects

Revoke an inherited project role from user on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/project_user_role_inherited_to_projects

Normal response codes: 204

Request

Name	In	Type	Description
project_id	path	string	The project ID.
role_id	path	string	The role ID.
user_id	path	string	The user ID.

DELETE

/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects

Revoke an inherited project role from group on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/ext/OS-INHERIT/1.0/rel/project_group_role_inherited_to_projects

Normal response codes: 204

Request

Name	In	Type	Description
group_id	path	string	The group ID.
project_id	path	string	The project ID.
role_id	path	string	The role ID.

GET

/v3/role_assignments

List role assignments

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/role_assignments

Optional query parameters:

Name	In	Type	Description
effective (Optional)	query	key-only (no value required)	Returns the effective assignments, including any assignments gained by virtue of group membership.
include_names (Optional)	query	boolean	If set to true, then the names of any entities returned will be include as well as their IDs. Any value other than 0 (including no value) will be interpreted as true. New in version 3.6
include_subtree (Optional)	query	boolean	If set to true, then relevant assignments in the project hierarchy below the project specified in the <code>scope.project_id</code> query parameter are also included in the response. Any value other than 0 (including no value) for <code>include_subtree</code> will be interpreted as true. New in version 3.6
group_id (Optional)	query	string	Filters the response by a group ID.
role_id (Optional)	query	string	Filters the response by a role ID.
scope.domain.id (Optional)	query	string	Filters the response by a domain ID.
scope.OS-INHERIT:inherited_to (Optional)	query	string	Filters based on role assignments that are inherited. The only value of <code>inherited_to</code> that is currently supported is <code>projects</code> .
scope.project.id (Optional)	query	string	Filters the response by a project ID.
user_id (Optional)	query	string	Filters the response by a user ID.

Get a list of role assignments.

If no query parameters are specified, then this API will return a list of all role assignments.

```
{
  "role_assignments": [
    {
      "links": {
        "assignment": "http://example.com/identity/v3/domains/161718/users/313233/roles/123456"
      },
      "role": {
        "id": "123456"
      },
      "scope": {
        "domain": {
          "id": "161718"
        }
      }
    },
  ],
}
```

```

        "user": {
          "id": "313233"
        }
      },
      {
        "group": {
          "id": "101112"
        },
        "links": {
          "assignment": "http://example.com/identity/v3/projects/456789/groups/101112/roles/123456"
        },
        "role": {
          "id": "123456"
        },
        "scope": {
          "project": {
            "id": "456789"
          }
        }
      }
    ],
    "links": {
      "self": "http://example.com/identity/v3/role_assignments",
      "previous": null,
      "next": null
    }
  }
}

```

Since this list is likely to be very long, this API would typically always be used with one of more of the filter queries. Some typical examples are:

GET /v3/role_assignments?user.id={user_id} would list all role assignments involving the specified user.

GET /v3/role_assignments?scope.project.id={project_id} would list all role assignments involving the specified project.

It is also possible to list all role assignments within a tree of projects: GET /v3/role_assignments?scope.project.id={project_id}?include_subtree=true would list all role assignments involving the specified project and all sub-projects. include_subtree=true can only be specified in conjunction with scope.project.id, specifying it without this will result in an HTTP 400 Bad Request being returned.

Each role assignment entity in the collection contains a link to the assignment that gave rise to this entity.

The scope section in the list response is extended to allow the representation of role assignments that are inherited to projects.

```

{
  "role_assignments": [
    {
      "links": {
        "assignment": "http://example.com/identity/v3/OS-INHERIT/domains/161718/users/313233/roles/123456/inherited_to_projects"
      },
      "role": {
        "id": "123456"
      },
      "scope": {
        "domain": {
          "id": "161718"
        },
        "OS-INHERIT:inherited_to": "projects"
      },
      "user": {
        "id": "313233"
      }
    },
    {
      "group": {
        "id": "101112-"
      },
      "links": {
        "assignment": "http://example.com/identity/v3/projects/456789/groups/101112/roles/123456"
      },
      "role": {
        "id": "123456"
      },
      "scope": {
        "project": {
          "id": "456789"
        }
      }
    }
  ]
}

```

```

    }
  },
  "links": {
    "self": "http://example.com/identity/v3/role_assignments",
    "previous": null,
    "next": null
  }
}

```

The query filter `scope.OS-INHERIT:inherited_to` can be used to filter based on role assignments that are inherited. The only value of `scope.OS-INHERIT:inherited_to` that is currently supported is `projects`, indicating that this role is inherited to all projects of the owning domain or parent project.

If the query parameter `effective` is specified, rather than simply returning a list of role assignments that have been made, the API returns a list of effective assignments at the user, project and domain level, having allowed for the effects of group membership, role inference rules as well as inheritance from the parent domain or project. Since the effects of group membership have already been allowed for, the group role assignment entities themselves will not be returned in the collection. Likewise, since the effects of inheritance have already been allowed for, the role assignment entities themselves that specify the inheritance will also not be returned in the collection. This represents the effective role assignments that would be included in a scoped token. The same set of query parameters can also be used in combination with the `effective` parameter.

For example:

GET `/v3/role_assignments?user.id={user_id}&effective` would, in other words, answer the question “what can this user actually do?”.

GET `/v3/role_assignments?user.id={user_id}&scope.project.id={project_id}&effective` would return the equivalent set of role assignments that would be included in the token response of a project scoped token.

An example response for an API call with the query parameter `effective` specified is given below:

```

{
  "role_assignments": [
    {
      "links": {
        "assignment": "http://example.com/identity/v3/domains/161718/users/313233/roles/123456"
      },
      "role": {
        "id": "123456"
      },
      "scope": {
        "domain": {
          "id": "161718"
        }
      },
      "user": {
        "id": "313233"
      }
    },
    {
      "links": {
        "assignment": "http://example.com/identity/v3/projects/456789/groups/101112/roles/123456",
        "membership": "http://example.com/identity/v3/groups/101112/users/313233"
      },
      "role": {
        "id": "123456"
      },
      "scope": {
        "project": {
          "id": "456789"
        }
      },
      "user": {
        "id": "313234"
      }
    }
  ],
  "links": {
    "self": "http://example.com/identity/v3/role_assignments?effective",
    "previous": null,
    "next": null
  }
}

```

```
}
```

The entity `links` section of a response using the `effective` query parameter also contains, for entities that are included by virtue of group membership, a url that can be used to access the membership of the group.

If the query parameter `include_names` is specified, rather than simply returning the entity IDs in the role assignments, the collection will additionally include the names of the entities. For example:

GET `/v3/role_assignments?`

`user.id={user_id}&effective&include_names=true` would return:

```
{
  "role_assignments": [
    {
      "links": {
        "assignment": "http://example.com/identity/v3/domains/161718/users/313233/roles/123456"
      },
      "role": {
        "domain": {
          "id": "161718",
          "name": "Default"
        },
        "id": "123456",
        "name": "admin"
      },
      "scope": {
        "domain": {
          "id": "161718",
          "name": "Default"
        }
      },
      "user": {
        "domain": {
          "id": "161718",
          "name": "Default"
        },
        "id": "313233",
        "name": "admin"
      }
    },
    {
      "links": {
        "assignment": "http://example.com/identity/v3/projects/456789/groups/101112/roles/123456",
        "membership": "http://example.com/identity/v3/groups/101112/users/313233"
      },
      "role": {
        "domain": {
          "id": "161718",
          "name": "Default"
        },
        "id": "123456",
        "name": "admin"
      },
      "scope": {
        "project": {
          "domain": {
            "id": "161718",
            "name": "Default"
          },
          "id": "456789",
          "name": "admin"
        }
      },
      "user": {
        "domain": {
          "id": "161718",
          "name": "Default"
        },
        "id": "313233",
        "name": "admin"
      }
    }
  ],
  "links": {
    "self": "http://example.com/identity/v3/role_assignments?effective&include_names=true",
    "previous": null,
    "next": null
  }
}
```

Normal response codes: 200

Error response codes: 400, 401, 403, 404, 405, 413, 503

OS-PKI API

GET

/v3/auth/tokens/OS-PKI/revoked

List revoked tokens

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/tokens/OS-PKI/revoked>

Lists revoked PKI tokens.

Normal response codes: 200

Error response codes: 413,415,405,404,403,401,400,503,409

Response

Name	In	Type	Description
signed	body	string	List of expired PKI tokens, signed by the cryptographic message syntax (CMS).

Response Example

```
{
  "signed": "-----BEGIN
CMS-----\nMIICGwYJKoZIhvcNAQcCoIICDDCCAggCAQExDTALBgUghkgBZQMEAgEwawYJKoZI\nnhvcNAQcBoF4EXHsicmV2b2t1ZCI6Ift7ImV4cGlyZXMiO\niAiMjAxNC0xMjQwMlQx\nnNzowMDowOVoiLCAiaWQiOiAiODhiMjRmOTI5OTk0NGU1ZjhkODE0MDNjYzMyY2M5\nnMmUifV19MYIBhTCCAYECAQEwXDBXMQswCQYDVQQGEWJVUzEOMAwGA1UECAwFVW5z\nnZlXQxZjAMBgNVBACMBVUuc2V0MQ4wDAYDVQQKDAVbnNldDEYMBYGA1UEAwwPd3d3\nnLmV4YW1wbGUuY29tAgEBMASGCWCGSFAwQCATANBgkqhkiG9w0BAQEFAASCAQA3\nnc8EI58ZxtqkyuUwqLPJZdB5v70u978w22Yk0sgL5ruUpQiwDhdgvL/sxqd70Pqi7\nnZZV3N+io+z1m4uAiSbriumv7HOEnIUEAUhK4G0kw5kAAg4j50c00mdiqd75k0j\n/nJPoRCXa8ieb0X87zhgfIq7ze/HZ7E2Lo020us3AEzmg1Nv023qgGcsSGPAUIHWN5\nn1oonPtgtziwVbmS2gs3Z9JB73mxEBviCX4CZEU/sNpchAzI/53tscKlqlzv+GBcm\nn1dYP3hEZn3twFRi9zos4hTwFkUivn6D3qqQB684sVrvK1z0CIq0KVGGYVSy/FQLE\nnWwQ5u58ZD8ohaJPu2Q6l\nn-----END CMS-----\n"
}
```

Policies

A policy is an arbitrarily serialized policy engine rule set to be consumed by a remote service.

You encode policy rule sets into a blob that remote services can consume. To do so, set `type` to `application/json` and specify policy rules as JSON strings in a `blob`. For example:

```
{
  "blob": {
    "foobar_user": [
      "role:compute-user"
    ]
  }
}
```

POST

/v3/policies

Create policy

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/policies>

Creates a policy.

Normal response codes: 201

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
policy	body	object	A <code>policy</code> object.
type	body	string	The MIME media type of the serialized policy blob.
blob	body	string	The policy rule set itself, as a serialized blob.

Request Example

```
{
  "policy": {
    "blob": "{\"foobar_user': 'role:compute-user'}",
    "type": "application/json"
  }
}
```

Response Parameters

Name	In	Type	Description
links	body	object	The links for the <code>policy</code> resource.
blob	body	string	The policy rule set itself, as a serialized blob.
policy	body	object	A <code>policy</code> object.
type	body	string	The MIME media type of the serialized policy blob.
id	body	string	The policy ID.

GET

/v3/policies

List policies

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/policies>

Lists policies.

Normal response codes: 200

Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
type (Optional)	query	string	Filters the response by a MIME media type for the serialized policy blob. For example, application/json.

Response Parameters

Name	In	Type	Description
links	body	object	The links for the policy resource.
blob	body	object	The policy rule itself, as a serialized blob.
policies	body	array	A policies object.
type	body	string	The MIME media type of the serialized policy blob.
id	body	string	The policy ID.

Response Example

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://example.com/identity/v3/policies"
  },
  "policies": [
    {
      "blob": {
        "foobar_user": [
          "role:compute-user"
        ]
      },
      "id": "717273",
      "links": {
        "self": "http://example.com/identity/v3/policies/717273"
      },
      "type": "application/json"
    },
    {
      "blob": {
        "foobar_user": [
          "role:compute-user"
        ]
      },
      "id": "717274",
      "links": {
        "self": "http://example.com/identity/v3/policies/717274"
      },
      "type": "application/json"
    }
  ]
}
```

GET

/v3/policies/{policy_id}

Show policy details

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/policy>

Shows details for a policy.

Normal response codes: 200

Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
policy_id	path	string	The policy ID.

Response Parameters

Name	In	Type	Description
links	body	object	The links for the <code>policy</code> resource.
blob	body	object	The policy rule itself, as a serialized blob.
policy	body	object	A <code>policy</code> object.
type	body	string	The MIME media type of the serialized policy blob.
id	body	string	The policy ID.

Response Example

```
{
  "policy": {
    "blob": {
      "foobar_user": [
        "role:compute-user"
      ]
    },
    "id": "717273",
    "links": {
      "self": "http://example.com/identity/v3/policies/717273"
    },
    "type": "application/json"
  }
}
```

PATCH

/v3/policies/{policy_id}

Update policy

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/policy>

Updates a policy.

Normal response codes: 200

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
policy_id	path	string	The policy ID.
policy	body	object	A <code>policy</code> object.
type	body	string	The MIME media type of the serialized policy blob.
blob	body	object	The policy rule itself, as a serialized blob.

Request Example

```
{
  "policy": {
    "blob": {
      "foobar_user": [
        "role:compute-user"
      ]
    },
    "type": "application/json"
  }
}
```

Response Parameters

Name	In	Type	Description
links	body	object	The links for the <code>policy</code> resource.
blob	body	object	The policy rule itself, as a serialized blob.
policy	body	object	A <code>policy</code> object.
type	body	string	The MIME media type of the serialized policy blob.
id	body	string	The policy ID.

Response Example

```
{
  "policy": {
    "blob": {
      "foobar_user": [
        "role:compute-user"
      ]
    },
    "id": "717273",
    "links": {
      "self": "http://example.com/identity/v3/policies/717273"
    },
    "type": "application/json"
  }
}
```

DELETE

/v3/policies/{policy_id}

Delete policy

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/policy>

Deletes a policy.

Normal response codes: 204

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
policy_id	path	string	The policy ID.

Projects

A project is the base unit of resource ownership. Resources are owned by a specific project. A project is owned by a specific domain.

(Since Identity API v3.4) You can create a hierarchy of projects by setting a `parent_id` when you create a project. All projects in a hierarchy must be owned by the same domain.

(Since Identity API v3.6) Projects may, in addition to acting as containers for OpenStack resources, act as a domain (by setting the attribute `is_domain` to `true`), in which case it provides a namespace in which users, groups and other projects can be created. In fact, a domain created using the `POST /domains` API will actually be represented as a project with `is_domain` set to `true` with no parent (`parent_id` is null).

Given this, all projects are considered part of a project hierarchy. Projects created in a domain prior to v3.6 are represented as a two-level hierarchy, with a project that has `is_domain` set to `true` as the root and all other projects referencing the root as their parent.

A project acting as a domain can potentially also act as a container for OpenStack resources, although this depends on whether the policy rule for the relevant resource creation allows this.

Note

A project's name must be unique within a domain and no more than 64 characters. A project's name must be able to be sent within valid JSON, which could be any UTF-8 character. However, this is constrained to the given backend where project names are stored. For instance, MySQL's restrictions states that UTF-8 support is constrained to the characters in the Basic Multilingual Plane (BMP). Supplementary characters are not permitted. Note that this last restriction is generally true for all names within resources of the Identity API.

GET

/v3/projects

List projects

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/projects>

Lists projects.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Request Parameters

Name	In	Type	Description
domain_id (Optional)	query	string	Filters the response by a domain ID.
enabled (Optional)	query	boolean	If set to true, then only enabled projects will be returned. Any value other than 0 (including no value) will be interpreted as true.
is_domain (Optional)	query	boolean	If this is specified as true, then only projects acting as a domain are included. Otherwise, only projects that are not acting as a domain are included. New in version 3.6
name (Optional)	query	string	Filters the response by a project name.
parent_id (Optional)	query	string	Filters the response by a parent ID. New in version 3.4

Response Parameters

Name	In	Type	Description
links	body	object	The link to the collection of resources.
projects	body	array	A list of project objects, each containing:
is_domain	body	boolean	Indicates whether the project also acts as a domain. If set to true, this project acts as both a

Name	In	Type	Description
			<p>project and domain. As a domain, the project provides a name space in which you can create users, groups, and other projects. If set to <code>false</code>, this project behaves as a regular project that contains only resources.</p> <p>New in version 3.6</p>
description	body	string	The description of the project.
domain_id	body	string	The ID of the domain for the project.
enabled	body	boolean	If set to <code>true</code> , project is enabled. If set to <code>false</code> , project is disabled.
id	body	string	The ID for the project.
links	body	object	The link to the resources in question.
name	body	string	The name of the project.
parent_id	body	string	<p>The ID of the parent for the project.</p> <p>New in version 3.4</p>

Response Example

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://example.com/identity/v3/projects"
  },
  "projects": [
    {
      "is_domain": false,
      "description": null,
      "domain_id": "default",
      "enabled": true,
      "id": "0c4e939acacf4376bdcd1129f1a054ad",
      "links": {
        "self": "http://example.com/identity/v3/projects/0c4e939acacf4376bdcd1129f1a054ad"
      },
      "name": "admin",
      "parent_id": null
    },
    {
      "is_domain": false,
      "description": null,
      "domain_id": "default",
      "enabled": true,
      "id": "0cbd49cbf76d405d9c86562e1d579bd3",
      "links": {
        "self": "http://example.com/identity/v3/projects/0cbd49cbf76d405d9c86562e1d579bd3"
      },
      "name": "demo",
      "parent_id": null
    },
    {
      "is_domain": false,
      "description": null,
      "domain_id": "default",
      "enabled": true,
      "id": "2db68fed84324f29bb73130c6c2094fb",
      "links": {
        "self": "http://example.com/identity/v3/projects/2db68fed84324f29bb73130c6c2094fb"
      },
      "name": "swifttenanttest2",
    }
  ]
}
```

```

    "parent_id": null
  },
  {
    "is_domain": false,
    "description": null,
    "domain_id": "default",
    "enabled": true,
    "id": "3d594eb0f04741069dbbb521635b21c7",
    "links": {
      "self": "http://example.com/identity/v3/projects/3d594eb0f04741069dbbb521635b21c7"
    },
    "name": "service",
    "parent_id": null
  },
  {
    "is_domain": false,
    "description": null,
    "domain_id": "default",
    "enabled": true,
    "id": "43ebde53fc314b1c9ea2b8c5dc744927",
    "links": {
      "self": "http://example.com/identity/v3/projects/43ebde53fc314b1c9ea2b8c5dc744927"
    },
    "name": "swifttenanttest1",
    "parent_id": null
  },
  {
    "is_domain": false,
    "description": "",
    "domain_id": "1bc2169ca88e4cdaaba46d4c15390b65",
    "enabled": true,
    "id": "4b1eb781a47440acb8af9850103e537f",
    "links": {
      "self": "http://example.com/identity/v3/projects/4b1eb781a47440acb8af9850103e537f"
    },
    "name": "swifttenanttest4",
    "parent_id": null
  },
  {
    "is_domain": false,
    "description": null,
    "domain_id": "default",
    "enabled": true,
    "id": "5961c443439d4fcebe42643723755e9d",
    "links": {
      "self": "http://example.com/identity/v3/projects/5961c443439d4fcebe42643723755e9d"
    },
    "name": "invisible_to_admin",
    "parent_id": null
  },
  {
    "is_domain": false,
    "description": null,
    "domain_id": "default",
    "enabled": true,
    "id": "fdb8424c4e4f4c0ba32c52e2de3bd80e",
    "links": {
      "self": "http://example.com/identity/v3/projects/fdb8424c4e4f4c0ba32c52e2de3bd80e"
    },
    "name": "alt_demo",
    "parent_id": null
  }
]
}

```

POST

/v3/projects

Create project

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/projects>

Creates a project, where the project may act as a domain.

Response Codes

Success

Code	Reason
201 - Created	Resource was created and is ready to use.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
project	body	object	A <code>project</code> object, containing:
is_domain (Optional)	body	boolean	<p>Indicates whether the project also acts as a domain. If set to <code>true</code>, this project acts as both a project and domain. As a domain, the project provides a name space in which you can create users, groups, and other projects. If set to <code>false</code>, this project behaves as a regular project that contains only resources. Default is <code>false</code>. You cannot update this parameter after you create the project.</p> <p>New in version 3.6</p>
description (Optional)	body	string	The description of the project.
domain_id (Optional)	body	string	<p>The ID of the domain for the project.</p> <p>For projects acting as a domain, the <code>domain_id</code> must not be specified, it will be generated by the Identity service implementation.</p> <p>For regular projects (i.e. those not acting as a domain), if <code>domain_id</code> is not specified, but <code>parent_id</code> is specified, then the domain ID of the parent will be used. If neither <code>domain_id</code> or <code>parent_id</code> is specified, the Identity service implementation will default to the domain to which the client's token is scoped. If both <code>domain_id</code> and <code>parent_id</code> are specified, and they do not indicate the same domain, an <code>Bad Request (400)</code> will be returned.</p>
enabled (Optional)	body	boolean	If set to <code>true</code> , project is enabled. If set to <code>false</code> ,

Name	In	Type	Description
			project is disabled. The default is <code>true</code> .
name	body	string	The name of the project, which must be unique within the owning domain. A project can have the same name as its domain.
parent_id (Optional)	body	string	<p>The ID of the parent of the project.</p> <p>If specified on project creation, this places the project within a hierarchy and implicitly defines the owning domain, which will be the same domain as the parent specified. If <code>parent_id</code> is not specified and <code>is_domain</code> is <code>false</code>, then the project will use its owning domain as its parent. If <code>is_domain</code> is <code>true</code> (i.e. the project is acting as a domain), then <code>parent_id</code> must not be specified (or if it is, it must be <code>null</code>) since domains have no parents.</p> <p><code>parent_id</code> is immutable, and can't be updated after the project is created - hence a project cannot be moved within the hierarchy.</p> <p>New in version 3.4</p>

Request Examples

Sample for creating a regular project:

```
{
  "project": {
    "description": "My new project",
    "domain_id": "default",
    "enabled": true,
    "is_domain": false,
    "name": "myNewProject"
  }
}
```

Sample for creating a project that also acts as a domain:

```
{
  "project": {
    "description": "My new domain",
    "enabled": true,
    "is_domain": true,
    "name": "myNewDomain"
  }
}
```

Response Parameters

Name	In	Type	Description
project	body	object	A project object, containing:

Name	In	Type	Description
is_domain	body	boolean	Indicates whether the project also acts as a domain. If set to <code>true</code> , this project acts as both a project and domain. As a domain, the project provides a name space in which you can create users, groups, and other projects. If set to <code>false</code> , this project behaves as a regular project that contains only resources. New in version 3.6
description	body	string	The description of the project.
domain_id	body	string	The ID of the domain for the project.
enabled	body	boolean	If set to <code>true</code> , project is enabled. If set to <code>false</code> , project is disabled.
id	body	string	The ID for the project.
links	body	object	The link to the resources in question.
name	body	string	The name of the project.
parent_id	body	string	The ID of the parent for the project. New in version 3.4

GET

/v3/projects/{project_id}

Show project details

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/project>

Shows details for a project.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.

Code	Reason
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
project_id	path	string	The project ID.
parents_as_list (Optional)	query	key-only, no value expected	The parent hierarchy will be included as a list in the response. This list will contain the projects found by traversing up the hierarchy to the top-level project. New in version 3.4
subtree_as_list (Optional)	query	key-only, no value expected	The child hierarchy will be included as a list in the response. This list will contain the projects found by traversing down the hierarchy. New in version 3.4
parents_as_ids (Optional)	query	key-only, no value expected	The entire parent hierarchy will be included as nested dictionaries in the response. It will contain all projects ids found by traversing up the hierarchy to the top-level project. New in version 3.4
subtree_as_ids (Optional)	query	key-only, no value expected	The entire child hierarchy will be included as nested dictionaries in the response. It will contain all the projects ids found by traversing down the hierarchy. New in version 3.4

Response Parameters

Name	In	Type	Description
project	body	object	A project object, containing:
is_domain	body	boolean	Indicates whether the project also acts as a domain. If set to true, this project acts as both

Name	In	Type	Description
			<p>a project and domain. As a domain, the project provides a name space in which you can create users, groups, and other projects. If set to <code>false</code>, this project behaves as a regular project that contains only resources.</p> <p>New in version 3.6</p>
description	body	string	The description of the project.
domain_id	body	string	The ID of the domain for the project.
enabled	body	boolean	If set to <code>true</code> , project is enabled. If set to <code>false</code> , project is disabled.
id	body	string	The ID for the project.
links	body	object	The link to the resources in question.
name	body	string	The name of the project.
parent_id	body	string	<p>The ID of the parent for the project.</p> <p>New in version 3.4</p>

Response Example

```
{
  "project": {
    "is_domain": false,
    "description": null,
    "domain_id": "default",
    "enabled": true,
    "id": "0c4e939acacf4376bdcd1129f1a054ad",
    "links": {
      "self": "http://example.com/identity/v3/projects/0c4e939acacf4376bdcd1129f1a054ad"
    },
    "name": "admin",
    "parent_id": "default"
  }
}
```

Response Example with parents_as_list

```
{
  "project": {
    "domain_id": "1789d1",
    "enabled": true,
    "id": "263fd9",
    "links": {
      "self": "http://example.com/identity/v3/projects/263fd9"
    },
    "name": "Dev Group A",
    "parent_id": "183ab2",
    "parents": [
      {
        "project": {
```

```

        "domain_id": "1789d1",
        "enabled": true,
        "id": "183ab2",
        "links": {
            "self": "http://example.com/identity/v3/projects/183ab2"
        },
        "name": "Dev Group A Parent",
        "parent_id": null
    }
}
]
}

```

Response Example with subtree_as_list

```

{
  "project": {
    "domain_id": "1789d1",
    "enabled": true,
    "id": "263fd9",
    "links": {
      "self": "http://example.com/identity/v3/projects/263fd9"
    },
    "name": "Dev Group A",
    "parent_id": "183ab2",
    "subtree": [
      {
        "project": {
          "domain_id": "1789d1",
          "enabled": true,
          "id": "9n1jhb",
          "links": {
            "self": "http://example.com/identity/v3/projects/9n1jhb"
          },
          "name": "Dev Group A Child 1",
          "parent_id": "263fd9"
        }
      },
      {
        "project": {
          "domain_id": "1789d1",
          "enabled": true,
          "id": "4b6aa1",
          "links": {
            "self": "http://example.com/identity/v3/projects/4b6aa1"
          },
          "name": "Dev Group A Child 2",
          "parent_id": "263fd9"
        }
      },
      {
        "project": {
          "domain_id": "1789d1",
          "enabled": true,
          "id": "b76eq8",
          "links": {
            "self": "http://example.com/identity/v3/projects/b76xq8"
          },
          "name": "Dev Group A Grandchild",
          "parent_id": "4b6aa1"
        }
      }
    ]
  }
}

```

PATCH

/v3/projects/{project_id}

Update project

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/project>

Updates a project.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
project_id	path	string	The project ID.
project	body	object	A project object, containing:
is_domain (Optional)	body	boolean	Indicates whether the project also acts as a domain. If set to <code>true</code> , this project acts as both a project and domain. As a domain, the project provides a name space in which you can create users, groups, and other projects. If set to <code>false</code> , this project behaves as a regular project that contains only resources. Default is <code>false</code> . You cannot update this parameter after you create the project. New in version 3.6
description (Optional)	body	string	The description of the project.
domain_id (Optional)	body	string	The ID of the new domain for the project. The ability to change the domain of a project is now deprecated, and will be removed in subsequent release. It is already disabled by default in most Identity service implementations.
enabled (Optional)	body	boolean	If set to <code>true</code> , project is enabled. If set to <code>false</code> , project is disabled.
name (Optional)	body	string	The name of the project, which must be unique within the owning domain. A project can have the same

Name	In	Type	Description
			name as its domain.

Request Example

```
{
  "project": {
    "description": "My updated project",
    "name": "myUpdatedProject"
  }
}
```

Response Parameters

Name	In	Type	Description
project	body	object	A project object, containing:
is_domain	body	boolean	Indicates whether the project also acts as a domain. If set to <code>true</code> , this project acts as both a project and domain. As a domain, the project provides a name space in which you can create users, groups, and other projects. If set to <code>false</code> , this project behaves as a regular project that contains only resources. New in version 3.6
description	body	string	The description of the project.
domain_id	body	string	The ID of the domain for the project.
enabled	body	boolean	If set to <code>true</code> , project is enabled. If set to <code>false</code> , project is disabled.
id	body	string	The ID for the project.
name	body	string	The name of the project.

Response Example

```
{
  "project": {
    "description": "My updated project",
    "domain_id": null,
    "links": {
      "self": "http://example.com/identity/v3/projects/93ebbcc35335488b96ff9cd7d18cbb2e"
    },
    "enabled": true,
    "id": "93ebbcc35335488b96ff9cd7d18cbb2e",
    "is_domain": true,
    "name": "myUpdatedProject"
    "parent_id": null,
  }
}
```

DELETE

/v3/projects/{project_id}

Delete project

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/project>

Deletes a project.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
Code	Policy does not allow current user to do this operation.
204 - No Content	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
project_id	path	string	The project ID.

Regions

A region is a general division of an OpenStack deployment. You can associate zero or more sub-regions with a region to create a tree- like structured hierarchy.

Although a region does not have a geographical connotation, a deployment can use a geographical name for a region ID, such as `us-east`.

You can list, create, update, show details for, and delete regions.

GET

`/v3/regions/{region_id}`

Show region details

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/project>

identity/3/rel/regions

Shows details for a region, by ID.

Normal response codes: 200

Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
region_id	path	string	The region ID.

Response Parameters

Name	In	Type	Description
region	body	object	A <code>region</code> object, containing the following:
description	body	string	The region description.
id	body	string	The ID for the region.
links	body	object	The links for the <code>region</code> resource.
parent_region_id	body	string	To make this region a child of another region, set this parameter to the ID of the parent region.

Response Example

```
{
  "region": {
    "description": "My subregion 3",
    "id": "RegionThree",
    "links": {
      "self": "http://example.com/identity/v3/regions/RegionThree"
    },
    "parent_region_id": "RegionOne"
  }
}
```

PATCH

/v3/regions/{region_id}

Update region

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/region>

Updates a region.

You can update the description or parent region ID for a region. You cannot update the region ID.

The following error might occur:

- Not Found (404). The parent region ID does not exist.

Normal response codes: 200

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
region_id	path	string	The region ID.
region	body	object	A region object, containing the following:
description (Optional)	body	string	The region description.
parent_region_id (Optional)	body	string	To make this region a child of another region, set this parameter to the ID of the parent region.

Request Example

```
{
  "region": {
    "description": "My subregion 3"
  }
}
```

Response Parameters

Name	In	Type	Description
region	body	object	A region object, containing the following:
description	body	string	The region description.
id	body	string	The ID for the region.
links	body	object	The links for the region resource.
parent_region_id	body	string	To make this region a child of another region, set this parameter to the ID of the parent region.

Response Example

```
{
  "region": {
    "parent_region_id": "RegionOne",
    "id": "RegionThree",
    "links": {
      "self": "http://example.com/identity/v3/regions/RegionThree"
    },
    "description": "My subregion 3"
  }
}
```

```
}  
}
```

DELETE

/v3/regions/{region_id}

Delete region

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/region>

Deletes a region.

The following error might occur:

- **Conflict (409)**. The region cannot be deleted because it has child regions.

Normal response codes: 204

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
region_id	path	string	The region ID.

GET

/v3/regions

List regions

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/regions>

Lists regions.

Normal response codes: 200

Error response codes: 413,405,404,403,401,400,503

Request

Name	In	Type	Description
parent_region_id (Optional)	query	string	Filters the response by a parent region, by ID.

Response Parameters

Name	In	Type	Description
regions	body	array	A list of <code>region</code> object, each containing the following:

Name	In	Type	Description
description	body	string	The region description.
id	body	string	The ID for the region.
links	body	object	The links for the region resource.
parent_region_id	body	string	To make this region a child of another region, set this parameter to the ID of the parent region.

Response Example

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://example.com/identity/v3/regions"
  },
  "regions": [
    {
      "description": "",
      "id": "RegionOne",
      "links": {
        "self": "http://example.com/identity/v3/regions/RegionOne"
      },
      "parent_region_id": null
    }
  ]
}
```

POST

/v3/regions

Create region

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/regions>

Creates a region.

When you create the region, you can optionally specify a region ID. If you include characters in the region ID that are not allowed in a URI, you must URL-encode the ID. If you omit an ID, the API assigns an ID to the region.

The following errors might occur:

- **Not Found (404)**. The parent region ID does not exist.
- **Conflict (409)**. The parent region ID would form a circular relationship.
- **Conflict (409)**. The user-defined region ID is not unique to the OpenStack deployment.

Normal response codes: 201

Error response codes: 413,415,405,404,403,401,400,503,409

Request

Name	In	Type	Description
region	body	object	A <code>region</code> object, containing the following:
description (Optional)	body	string	The region description.
id (Optional)	body	string	The ID for the region.
parent_region_id (Optional)	body	string	To make this region a child of another region, set this parameter to the ID of the parent region.

Request Example

```
{
  "region": {
    "description": "My subregion",
    "id": "RegionOneSubRegion",
    "parent_region_id": "RegionOne"
  }
}
```

Response Parameters

Name	In	Type	Description
region	body	object	A <code>region</code> object, containing the following:
description	body	string	The region description.
id	body	string	The ID for the region.
links	body	object	The links for the <code>region</code> resource.
parent_region_id	body	string	To make this region a child of another region, set this parameter to the ID of the parent region.

Roles

OpenStack services typically determine whether a user's API request should be allowed using Role Based Access Control (RBAC). For OpenStack this means the service compares the roles that user has on the project (as indicated by the roles in the token), against the roles required for the API in question (as defined in the service's policy file). A user obtains roles on a project by having these assigned to them via the Identity service API.

Roles must initially be created as entities via the Identity services API and, once created, can then be assigned. You can assign roles to a user or group on a project, including projects owned by other domains. You can also assign roles to a user or group on a domain, although this is only currently relevant for using a domain scoped token to execute domain-level Identity service API requests.

The creation, checking and deletion of role assignments is done with each of the attributes being specified

in the URL. For example to assign a role to a user on a project:

```
PUT /v3/projects/{project_id}/users/{user_id}/roles/{role_id}
```

You can also list roles assigned to a specified domain, project, or user using this form of API, however a more generalized API for list assignments is provided where query parameters are used to filter the set of assignments returned in the collection. For example:

- List role assignments for the specified user:

```
GET /role_assignments?user.id={user_id}
```

- List role assignments for the specified project:

```
GET /role_assignments?scope.project.id={project_id}
```

Since Identity API v3.6, you can also list all role assignments within a tree of projects, for example the following would list all role assignments for a specified project and its sub-projects:

```
GET /role_assignments?scope.project.id={project_id}&include_subtree=true
```

If you specify `include_subtree=true`, you must also specify the `scope.project.id`. Otherwise, this call returns the **Bad Request (400)** response code.

Each role assignment entity in the collection contains a link to the assignment that created the entity.

As mentioned earlier, role assignments can be made to a user or a group on a particular project or domain. A user who is a member of a group that has a role assignment, will also be treated as having that role assignment by virtue of their group membership. The *effective* role assignments of a user (on a given project or domain) therefore consists of any direct assignments they have, plus any they gain by virtue of membership of groups that also have assignments on the given project or domain. This set of effective role assignments is what is placed in the token for reference by services wishing to check policy. You can list the effective role assignments using the `effective` query parameter at the user, project, and domain level:

- Determine what a user can actually do:

```
GET /role_assignments?user.id={user_id}&effective
```

- Get the equivalent set of role assignments that are included in a project-scoped token response:

```
GET /role_assignments?user.id={user_id}&scope.project.id={project_id}&effective
```

When listing in effective mode, since the group assignments have been effectively expanded out into assignments for each user, the group role assignment entities themselves are not returned in the collection. However, in the response, the `links` entity section for each assignment gained by virtue of group membership will contain a URL that enables access to the membership of the group.

By default only the IDs of entities are returned in collections from the `role_assignment` API calls. The names of entities may also be returned, in addition to the IDs, by using the `include_names` query parameter on any of these calls, for example:

- List role assignments including names:

```
GET /role_assignments?include_names
```

GET

/v3/roles

List roles

Relationship: <https://docs.openstack.org/api/openstack-identity/3/re1/roles>

Lists roles.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Request Parameters

Name	In	Type	Description
name (Optional)	query	string	Filters the response by a role name.
domain_id (Optional)	query	string	Filters the response by a domain ID.

Response Parameters

Name	In	Type	Description
links	body	object	The link to the collection of resources.
roles	body	array	A list of <code>role</code> objects, each containing:
domain_id	body	string	The ID of the domain.
id	body	string	The role ID.
links	body	object	The link to the resources in question.
name	body	string	The role name.

Response Example

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://example.com/identity/v3/roles"
  },
  "roles": [
    {
      "id": "5318e65d75574c17bf5339d3df33a5a3",
      "links": {
        "self": "http://example.com/identity/v3/roles/5318e65d75574c17bf5339d3df33a5a3"
      },
      "name": "admin"
    },
    {
      "id": "642bcfc75c384fd181adf34d9b2df897",
      "links": {
        "self": "http://example.com/identity/v3/roles/642bcfc75c384fd181adf34d9b2df897"
      },
      "name": "anotherrole"
    },
    {
      "id": "779a76d74f544224a7ef8762ca0de627",
      "links": {
        "self": "http://example.com/identity/v3/roles/779a76d74f544224a7ef8762ca0de627"
      },
      "name": "Member"
    },
    {
      "id": "9fe2ff9ee4384b1894a90878d3e92bab",
      "links": {
        "self": "http://example.com/identity/v3/roles/9fe2ff9ee4384b1894a90878d3e92bab"
      },
      "name": "_member_"
    },
    {
      "id": "ba2dfba61c934ee89e3110de36273229",
      "links": {
        "self": "http://example.com/identity/v3/roles/ba2dfba61c934ee89e3110de36273229"
      },
      "name": "ResellerAdmin"
    },
    {
      "id": "f127b97616f24d3ebceb7be840210adc",
      "links": {
        "self": "http://example.com/identity/v3/roles/f127b97616f24d3ebceb7be840210adc"
      },
      "name": "service"
    }
  ]
}
```

POST

/v3/roles

Create role

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/roles>

Creates a role.

Response Codes

Success

Code	Reason
201 - Created	Resource was created and is ready to use.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
role	body	object	A role object, containing:
name	body	string	The role name.
domain_id	body	string	The ID of the domain.

Request Example

```
{
  "role": {
    "name": "developer"
  }
}
```

Request Example for Domain Specific Role

```
{
  "role": {
    "domain_id": "92e782c4988642d783a95f4a87c3fdd7",
    "name": "developer"
  }
}
```

Response Parameters

Name	In	Type	Description
role	body	object	A role object, containing:
domain_id	body	string	The ID of the domain.
id	body	string	The role ID.
links	body	object	The link to the resources in question.
name	body	string	The role name.

GET

/v3/roles/{role_id}

Show role details

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/role>

Shows details for a role.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
role_id	path	string	The role ID.

Response Parameters

Name	In	Type	Description
role	body	object	A role object, containing:
domain_id	body	string	The ID of the domain.
id	body	string	The role ID.
links	body	object	The link to the resources in question.
name	body	string	The role name.

Response Example

```
{
  "role": {
    "domain_id": "d07792fd66ac4ed881723ab9f1c9925f",
    "id": "1e443fa8cee3482a8a2b6954dd5c8f12",
    "links": {
```

```
        "self": "http://example.com/identity/v3/roles/1e443fa8cee3482a8a2b6954dd5c8f12"
    },
    "name": "Developer"
}
}
```

PATCH

/v3/roles/{role_id}

Update role

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/role>

Updates a role.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
role_id	path	string	The role ID.
role	body	object	A role object, containing:
name (Optional)	body	string	The new role name.

Request Example

```
{
  "role": {
    "name": "Developer"
  }
}
```

Response Parameters

Name	In	Type	Description
role	body	object	A role object, containing:
domain_id	body	string	The ID of the domain.
id	body	string	The role ID.
links	body	object	The link to the resources in question.
name	body	string	The role name.

Response Example

```
{
  "role": {
    "domain_id": "73748865fb964ded9e836d491d32dcfb",
    "id": "1e443fa8cee3482a8a2b6954dd5c8f12",
    "links": {
      "self": "http://example.com/identity/v3/roles/1e443fa8cee3482a8a2b6954dd5c8f12"
    },
    "name": "Developer"
  }
}
```

DELETE

/v3/roles/{role_id}

Delete role

Relationship: <https://docs.openstack.org/api/openstack-identity/3/re1/role>

Deletes a role.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
role_id	path	string	The role ID.

GET

/v3/domains/{domain_id}/groups/{group_id}/roles

List role assignments for group on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_group_roles

Lists role assignments for a group on a domain.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Request Parameters

Name	In	Type	Description
domain_id	path	string	The domain ID.
group_id	path	string	The group ID.

Response Example

```
{
  "roles": [
    {
      "id": "123456",
      "links": {
        "self": "http://example.com/identity/v3/roles/123456"
      },
      "name": "admin"
    },
    {
      "id": "123457",
      "links": {
        "self": "http://example.com/identity/v3/roles/123457"
      },
      "name": "manager"
    }
  ],
}
```

```
    "links": {  
      "self": "http://example.com/identity/v3/domains/161718/groups/101112/roles",  
      "previous": null,  
      "next": null  
    }  
  }  
}
```

The functionality of this request can also be achieved using the generalized list assignments API:

GET /role_assignments?group.id={group_id}&scope.domain.id={domain_id}

PUT

/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}

Assign role to group on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_group_role

Assigns a role to a group on a domain.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
domain_id	path	string	The domain ID.
group_id	path	string	The group ID.
role_id	path	string	The role ID.

HEAD

/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}

Check whether group has role assignment on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_group_role

Validates that a group has a role assignment on a domain.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
domain_id	path	string	The domain ID.
group_id	path	string	The group ID.
role_id	path	string	The role ID.

DELETE

/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}

Unassign role from group on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_group_role

Unassigns a role from a group on a domain.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
domain_id	path	string	The domain ID.
group_id	path	string	The group ID.
role_id	path	string	The role ID.

GET

/v3/domains/{domain_id}/users/{user_id}/roles

List role assignments for user on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_user_roles

Lists role assignments for a user on a domain.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Request Parameters

Name	In	Type	Description
domain_id	path	string	The domain ID.
user_id	path	string	The user ID.

Response Parameters

Name	In	Type	Description
roles	body	array	A list of role objects, each containing:
id	body	string	The role ID.
links	body	object	The link to the resources in question.
name	body	string	The role name.

Response Example

```
{
  "roles": [
    {
      "id": "123456",
      "links": {
        "self": "http://example.com/identity/v3/roles/123456"
      },
      "name": "admin"
    },
    {
      "id": "123457",
      "links": {
        "self": "http://example.com/identity/v3/roles/123457"
      },
      "name": "manager"
    }
  ],
  "links": {
    "self": "http://example.com/identity/v3/domains/161718/users/313233/roles",
    "previous": null,
    "next": null
  }
}
```

The functionality of this request can also be achieved using the generalized list assignments API:

GET /role_assignments?user.id={user_id}&scope.domain.id={domain_id}

PUT`/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}`

Assign role to user on domain

Relationship: <https://developer.openstack.org/api-ref/identity/v3/index.html#assign-role-to-user-on-domain>

Assigns a role to a user on a domain.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
domain_id	path	string	The domain ID.
user_id	path	string	The user ID.
role_id	path	string	The role ID.

HEAD`/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}`

Check whether user has role assignment on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_user_role

Validates that a user has a role assignment on a domain.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
domain_id	path	string	The domain ID.
user_id	path	string	The user ID.
role_id	path	string	The role ID.

DELETE

/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}

Unassigns role from user on domain

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/domain_user_role

Unassigns a role from a user on a domain.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
domain_id	path	string	The domain ID.
user_id	path	string	The user ID.
role_id	path	string	The role ID.

GET

/v3/projects/{project_id}/groups/{group_id}/roles

List role assignments for group on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/project_user_role

Lists role assignments for a group on a project.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before

Code	Reason
	making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Request Parameters

Name	In	Type	Description
project_id	path	string	The project ID.
group_id	path	string	The group ID.

Response Example

```
{
  "roles": [
    {
      "id": "123456",
      "links": {
        "self": "http://example.com/identity/v3/roles/123456"
      },
      "name": "admin"
    },
    {
      "id": "123457",
      "links": {
        "self": "http://example.com/identity/v3/roles/123457"
      },
      "name": "manager"
    }
  ],
  "links": {
    "self": "http://example.com/identity/v3/projects/456789/groups/101112/roles",
    "previous": null,
    "next": null
  }
}
```

The functionality of this request can also be achieved using the generalized list assignments API:

GET /role_assignments?group.id={group_id}&scope.project.id={project_id}

PUT

/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}

Assign role to group on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/project_group_role

Assigns a role to a group on a project.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
project_id	path	string	The project ID.
group_id	path	string	The group ID.
role_id	path	string	The role ID.

HEAD

/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}

Check whether group has role assignment on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/project_group_role

Validates that a group has a role assignment on a project.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user

Code	Reason
	to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
project_id	path	string	The project ID.
group_id	path	string	The group ID.
role_id	path	string	The role ID.

DELETE

/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}

Unassign role from group on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/project_group_role

Unassigns a role from a group on a project.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
project_id	path	string	The project ID.
group_id	path	string	The group ID.
role_id	path	string	The role ID.

GET

/v3/projects/{project_id}/users/{user_id}/roles

List role assignments for user on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/project_user_role

Lists role assignments for a user on a project.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Request Parameters

Name	In	Type	Description
project_id	path	string	The project ID.
user_id	path	string	The user ID.

Response Example

```
{
  "links": {
    "self": "http://example.com/identity/v3/projects/9e5a15e2c0dd42aab0990a463e839ac1/users/b964a9e51c0046a4a84d3f83a135a97c/roles",
    "previous": null,
    "next": null
  },
  "roles": [
    {
```

```
    "id": "3b5347fa7a144008ba57c0acea469cc3",
    "links": {
      "self": "http://example.com/identity/v3/roles/3b5347fa7a144008ba57c0acea469cc3"
    },
    "name": "admin"
  }
]
```

PUT

/v3/projects/{project_id}/users/{user_id}/roles/{role_id}

Assign role to user on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/re1/project_user_role

Assigns a role to a user on a project.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
project_id	path	string	The project ID.
user_id	path	string	The user ID.
role_id	path	string	The role ID.

HEAD

/v3/projects/{project_id}/users/{user_id}/roles/{role_id}

Check whether user has role assignment on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/project_user_role

Validates that a user has a role on a project.

Response Codes

Success

Code	Reason
201 - Created	Resource was created and is ready to use.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
project_id	path	string	The project ID.
user_id	path	string	The user ID.
role_id	path	string	The role ID.

DELETE

/v3/projects/{project_id}/users/{user_id}/roles/{role_id}

Unassign role from user on project

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/project_user_role

Unassigns a role from a user on a project.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
project_id	path	string	The project ID.
user_id	path	string	The user ID.
role_id	path	string	The role ID.

GET

/v3/roles/{prior_role_id}/implies

List implied (inference) roles for role

Lists implied (inference) roles for a role.

Relationship: <https://developer.openstack.org/api-ref/identity/v3/#list-implied-roles-for-role>

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
401 - Unauthorized	User must authenticate before making a request.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
prior_role_id	path	string	Role ID for a prior role.

Response Parameters

Name	In	Type	Description
role_inference	body	object	Role inference object that contains prior_role object and implies object.
prior_role	body	object	A prior role object.
implies	body	array	An array of implied role objects.
id	body	string	The role ID.
links	body	object	The link to the resources in question.
name	body	string	The role name.

Response Example

```
{
  "role_inference": {
    "prior_role": {
      "id": "42c764f0c19146728dbfe73a49cc35c3",
      "links": {
        "self": "http://example.com/identity/v3/roles/42c764f0c19146728dbfe73a49cc35c3"
      },
      "name": "prior role name"
    },
    "implies": [
      {
        "id": "066fbfc8b3e54fb68784c9e7e92ab8d7",
        "links": {
          "self": "http://example.com/identity/v3/roles/066fbfc8b3e54fb68784c9e7e92ab8d7"
        },
        "name": "implied role1 name"
      },
      {
        "id": "32a0df1cc22848aca3986adae9e0b9a0",
        "links": {
          "self": "http://example.com/identity/v3/roles/32a0df1cc22848aca3986adae9e0b9a0"
        },
        "name": "implied role2 name"
      }
    ],
    "links": {
      "self": "http://example.com/identity/v3/roles/42c764f0c19146728dbfe73a49cc35c3/implies"
    }
  }
}
```

PUT

/v3/roles/{prior_role_id}/implies/{implies_role_id}

Create role inference rule

Creates a role inference rule.

Relationship: <https://developer.openstack.org/api-ref/identity/v3/#create-role-inference-rule>

Response Codes

Success

Code	Reason
201 - Created	Resource was created and is ready to use.

Error

Code	Reason
401 - Unauthorized	User must authenticate before making a request.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
prior_role_id	path	string	Role ID for a prior role.
implies_role_id	path	string	Role ID for an implied role.

Response Parameters

Name	In	Type	Description
role_inference	body	object	Role inference object that contains prior_role object and implies object.
prior_role	body	object	A prior role object.
implies	body	object	An implied role object.
id	body	string	The role ID.
links	body	object	The link to the resources in question.
name	body	string	The role name.

Response Example

```
{
  "role_inference": {
    "prior_role": {
      "id": "7ceab6192ea34a548cc71b24f72e762c",
      "links": {
        "self": "http://example.com/identity/v3/roles/7ceab6192ea34a548cc71b24f72e762c"
      },
      "name": "prior role name"
    },
    "implies": {
      "id": "97e2f5d38bc94842bc3da818c16762ed",
      "links": {
        "self": "http://example.com/identity/v3/roles/97e2f5d38bc94842bc3da818c16762ed"
      },
      "name": "implied role name"
    }
  },
  "links": {
    "self": "http://example.com/identity/v3/roles/7ceab6192ea34a548cc71b24f72e762c/implies/97e2f5d38bc94842bc3da818c16762ed"
  }
}
```

GET

/v3/roles/{prior_role_id}/implies/{implies_role_id}

Get role inference rule

Gets a role inference rule.

Relationship: <https://developer.openstack.org/api-ref/identity/v3/#get-role-inference-rule>

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
401 - Unauthorized	User must authenticate before making a request.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
prior_role_id	path	string	Role ID for a prior role.
implies_role_id	path	string	Role ID for an implied role.

Response Parameters

Name	In	Type	Description
role_inference	body	object	Role inference object that contains prior_role object and implies object.
prior_role	body	object	A prior role object.
implies	body	object	An implied role object.
id	body	string	The role ID.
links	body	object	The link to the resources in question.
name	body	string	The role name.

Response Example

```
{
  "role_inference": {
    "prior_role": {
      "id": "7ceab6192ea34a548cc71b24f72e762c",
      "links": {
        "self": "http://example.com/identity/v3/roles/7ceab6192ea34a548cc71b24f72e762c"
      },
      "name": "prior role name"
    },
    "implies": {
      "id": "97e2f5d38bc94842bc3da818c16762ed",
      "links": {
        "self": "http://example.com/identity/v3/roles/97e2f5d38bc94842bc3da818c16762ed"
      },
      "name": "implied role name"
    }
  },
  "links": {
    "self":
"http://example.com/identity/v3/roles/7ceab6192ea34a548cc71b24f72e762c/implies/97e2f5d38bc94842bc3da818c16762ed"
  }
}
```

HEAD

/v3/roles/{prior_role_id}/implies/{implies_role_id}

Confirm role inference rule

Checks a role role inference rule.

Relationship: <https://developer.openstack.org/api-ref/identity/v3/#confirm-role-inference-rule>

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
401 - Unauthorized	User must authenticate before making a request.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
prior_role_id	path	string	Role ID for a prior role.
implies_role_id	path	string	Role ID for an implied role.

Response Example

Status: 204 No Content

DELETE

/v3/roles/{prior_role_id}/implies/{implies_role_id}

Delete role inference rule

Deletes a role inference rule.

Relationship: <https://developer.openstack.org/api-ref/identity/v3/#delete-role-inference-rule>

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
401 - Unauthorized	User must authenticate before making a request.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
prior_role_id	path	string	Role ID for a prior role.

Name	In	Type	Description
implies_role_id	path	string	Role ID for an implied role.

Response Example

Status: 204 No Content

GET

/v3/role_assignments

List role assignments

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/role_assignments

Lists role assignments.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Request Parameters

Name	In	Type	Description
effective (Optional)	query	key-only (no value required)	Returns the effective assignments, including any assignments gained by virtue of group membership.
include_names (Optional)	query	boolean	<p>If set to true, then the names of any entities returned will be include as well as their IDs. Any value other than 0 (including no value) will be interpreted as true.</p> <p>New in version 3.6</p>

Name	In	Type	Description
include_subtree (Optional)	query	boolean	If set to true, then relevant assignments in the project hierarchy below the project specified in the <code>scope.project_id</code> query parameter are also included in the response. Any value other than 0 (including no value) for <code>include_subtree</code> will be interpreted as true. New in version 3.6
group.id (Optional)	query	string	Filters the response by a group ID.
role.id (Optional)	query	string	Filters the response by a role ID.
scope.domain.id (Optional)	query	string	Filters the response by a domain ID.
scope.project.id (Optional)	query	string	Filters the response by a project ID.
user.id (Optional)	query	string	Filters the response by a user ID.

Response Parameters

Name	In	Type	Description
role_assignments	body	array	A list of <code>role_assignment</code> objects.

Response Example

```
{
  "role_assignments": [
    {
      "links": {
        "assignment": "http://example.com/identity/v3/domains/161718/users/313233/roles/123456"
      },
      "role": {
        "id": "123456"
      },
      "scope": {
        "domain": {
          "id": "161718"
        }
      },
      "user": {
        "id": "313233"
      }
    },
    {
      "group": {
        "id": "101112"
      },
      "links": {
        "assignment": "http://example.com/identity/v3/projects/456789/groups/101112/roles/123456"
      },
      "role": {
        "id": "123456"
      },
      "scope": {
        "project": {
          "id": "456789"
        }
      }
    }
  ],
  "links": {
    "self": "http://example.com/identity/v3/role_assignments",
    "previous": null,
    "next": null
  }
}
```

/v3/role_inferences

List all role inference rules

Lists all role inference rules.

Relationship: <https://developer.openstack.org/api-ref/identity/v3/#list-all-role-inference-rules>

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
401 - Unauthorized	User must authenticate before making a request.
404 - Not Found	The requested resource could not be found.

Response Parameters

Name	In	Type	Description
role_inferences	body	array	An array of role_inference object.
prior_role	body	object	A prior role object.
implies	body	object	An implied role object.
id	body	string	The role ID.
links	body	object	The link to the resources in question.
name	body	string	The role name.

Response Example

```
{
  "role_inferences": [
    {
      "prior_role": {
        "id": "1acd3c5aa0e246b9a7427d252160dcd1",
        "links": {
          "self": "http://example.com/identity/v3/roles/1acd3c5aa0e246b9a7427d252160dcd1"
        },
        "name": "prior role name"
      },
      "implies": [
        {
          "id": "3602510e2e1f499589f78a0724dcf614",
          "links": {
            "self": "http://example.com/identity/v3/roles/3602510e2e1f499589f78a0724dcf614"
          },
          "name": "implied role1 name"
        }
      ]
    }
  ]
}
```

```

    },
    {
      "id": "738289aeef684e73a987f7cf2ec6d925",
      "links": {
        "self": "http://example.com/identity/v3/roles/738289aeef684e73a987f7cf2ec6d925"
      },
      "name": "implied role2 name"
    }
  ]
},
{
  "prior_role": {
    "id": "bbf7a5098bb34407b7164eb6ff9f144e",
    "links": {
      "self": "http://example.com/identity/v3/roles/bbf7a5098bb34407b7164eb6ff9f144e"
    },
    "name": "prior role name"
  },
  "implies": [
    {
      "id": "872b20ad124c4c1bafaef2b1aae316ab",
      "links": {
        "self": "http://example.com/identity/v3/roles/872b20ad124c4c1bafaef2b1aae316ab"
      },
      "name": "implied role1 name"
    },
    {
      "id": "1d865b1b2da14cb7b05254677e5f36a2",
      "links": {
        "self": "http://example.com/identity/v3/roles/1d865b1b2da14cb7b05254677e5f36a2"
      },
      "name": "implied role2 name"
    }
  ]
}
],
"links": {
  "self": "http://example.com/identity/v3/role_inferences"
}
}

```

Service catalog and endpoints

A service is an OpenStack web service that you can access through a URL, i.e. an endpoint.

A service catalog lists the services that are available to the caller based upon the current authorization.

You can create, list, show details for, update, and delete services. When you create or update a service, you can enable the service, which causes it and its endpoints to appear in the service catalog.

You can create, list, show details for, update, and delete endpoints.

GET
/v3/services

List services

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/services>

Lists all services.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Request Parameters

Name	In	Type	Description
type (Optional)	query	string	Filters the response by a service type. A valid value is <code>compute</code> , <code>ec2</code> , <code>identity</code> , <code>image</code> , <code>network</code> , or <code>volume</code> .

Response Parameters

Name	In	Type	Description
name	body	string	The service name.
links	body	object	The links for the service resource.
enabled	body	boolean	Defines whether the service and its endpoints appear in the service catalog: - <code>false</code> . The service and its endpoints do not appear in the service catalog. - <code>true</code> . The service and its endpoints appear in the service catalog.
services	body	array	A list of service object.
type	body	string	The service type, which describes the API implemented by the service. Value is <code>compute</code> , <code>ec2</code> , <code>identity</code> , <code>image</code> , <code>network</code> , or <code>volume</code> .
id	body	string	The UUID of the service to which the endpoint belongs.
description	body	string	The service description.

Response Example

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://example.com/identity/v3/services"
  },
}
```

```
"services": [
  {
    "description": "Nova Compute Service",
    "enabled": true,
    "id": "1999c3a858c7408fb586817620695098",
    "links": {
      "self": "http://example.com/identity/v3/services/1999c3a858c7408fb586817620695098"
    },
    "name": "nova",
    "type": "compute"
  },
  {
    "description": "Cinder Volume Service V2",
    "enabled": true,
    "id": "39216610e75547f1883037e11976fc0f",
    "links": {
      "self": "http://example.com/identity/v3/services/39216610e75547f1883037e11976fc0f"
    },
    "name": "cinderv2",
    "type": "volumev2"
  },
  {
    "description": "Neutron Service",
    "enabled": true,
    "id": "4fe41a27de3341af9100123f765eac0d",
    "links": {
      "self": "http://example.com/identity/v3/services/4fe41a27de3341af9100123f765eac0d"
    },
    "name": "neutron",
    "type": "network"
  },
  {
    "description": "EC2 Compatibility Layer",
    "enabled": true,
    "id": "61d3d05bdd1449f18923c83f52a4d762",
    "links": {
      "self": "http://example.com/identity/v3/services/61d3d05bdd1449f18923c83f52a4d762"
    },
    "name": "ec2",
    "type": "ec2"
  },
  {
    "description": "Glance Image Service",
    "enabled": true,
    "id": "69afa3d57d1948ea988beeb252bbaa5d",
    "links": {
      "self": "http://example.com/identity/v3/services/69afa3d57d1948ea988beeb252bbaa5d"
    },
    "name": "glance",
    "type": "image"
  },
  {
    "description": "Nova Compute Service V2.1",
    "enabled": true,
    "id": "79b691ee7be649d9bf8613efc0960206",
    "links": {
      "self": "http://example.com/identity/v3/services/79b691ee7be649d9bf8613efc0960206"
    },
    "name": "novav21",
    "type": "computev21"
  },
  {
    "description": "Swift Service",
    "enabled": true,
    "id": "92419b70ebe64c6c873bd20b14360e6b",
    "links": {
      "self": "http://example.com/identity/v3/services/92419b70ebe64c6c873bd20b14360e6b"
    },
    "name": "swift",
    "type": "object-store"
  },
  {
    "description": "Keystone Identity Service",
    "enabled": true,
    "id": "b8f8454fc07b46b781204d2a436f9d1c",
    "links": {
      "self": "http://example.com/identity/v3/services/b8f8454fc07b46b781204d2a436f9d1c"
    },
    "name": "keystone",
    "type": "identity"
  },
  {
    "description": "Cinder Volume Service",
    "enabled": true,
    "id": "cdda3bea0742407f95e70f4758f46558",
    "links": {

```

```
        "self": "http://example.com/identity/v3/services/cdda3bea0742407f95e70f4758f46558"
      },
      "name": "cinder",
      "type": "volume"
    }
  ]
}
```

POST

/v3/services

Create service

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/services>

Creates a service.

Response Codes

Success

Code	Reason
201 - Created	Resource was created and is ready to use.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
description	body	string	The service description.
service	body	object	A service object.
enabled	body	boolean	Defines whether the service and its endpoints appear in the service catalog: - <code>false</code> . The service and its endpoints do not appear in the service catalog. - <code>true</code> . The service and its endpoints appear in the service catalog.
type	body	string	The service type, which describes

Name	In	Type	Description
			the API implemented by the service. Value is <code>compute</code> , <code>ec2</code> , <code>identity</code> , <code>image</code> , <code>network</code> , or <code>volume</code> .
name	body	string	The service name.

Request Example

```
{
  "service": {
    "type": "compute",
    "name": "compute2",
    "description": "Compute service 2"
  }
}
```

Response Parameters

Name	In	Type	Description
name	body	string	The service name.
service	body	object	A <code>service</code> object.
links	body	object	The links for the <code>service</code> resource.
type	body	string	The service type, which describes the API implemented by the service. Value is <code>compute</code> , <code>ec2</code> , <code>identity</code> , <code>image</code> , <code>network</code> , or <code>volume</code> .
id	body	string	The UUID of the service to which the endpoint belongs.
description	body	string	The service description.

GET

/v3/services/{service_id}

Show service details

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/service>

Shows details for a service.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
service_id	path	string	The service ID.

Response Parameters

Name	In	Type	Description
name	body	string	The service name.
service	body	object	A service object.
links	body	object	The links for the <code>service</code> resource.
type	body	string	The service type, which describes the API implemented by the service. Value is <code>compute</code> , <code>ec2</code> , <code>identity</code> , <code>image</code> , <code>network</code> , or <code>volume</code> .
id	body	string	The UUID of the service to which the endpoint belongs.
description	body	string	The service description.

Response Example

```
{
  "service": {
    "description": "Keystone Identity Service",
    "enabled": true,
    "id": "686766",
    "links": {
      "self": "http://example.com/identity/v3/services/686766"
    },
    "name": "keystone",
```

```
    "type": "identity"
  }
}
```

PATCH

/v3/services/{service_id}

Update service

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/services>

Updates a service.

The request body is the same as the create service request body, except that you include only those attributes that you want to update.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
type	body	string	The service type, which describes the API implemented by the service. Value is <code>compute</code> , <code>ec2</code> , <code>identity</code> , <code>image</code> , <code>network</code> , or <code>volume</code> .
enabled (Optional)	body	boolean	Defines whether the service and its endpoints appear in the service catalog: - <code>false</code> . The service and its endpoints do not appear in the service catalog. - <code>true</code> . The service and its endpoints appear in the service catalog. Default is <code>true</code> .

Name	In	Type	Description
description (Optional)	body	string	The service description.
service	body	object	A service object.
name	body	string	The service name.
service_id	path	string	The service ID.

Request Example

```
{
  "service": {
    "description": "Block Storage Service V2"
  }
}
```

Response Parameters

Name	In	Type	Description
name	body	string	The service name.
service	body	object	A service object.
links	body	object	The links for the service resource.
type	body	string	The service type, which describes the API implemented by the service. Value is compute, ec2, identity, image, network, or volume.
id	body	string	The UUID of the service to which the endpoint belongs.
description	body	string	The service description.

Response Example

```
{
  "service": {
    "name": "cinderv2",
    "links": {
      "self": "http://example.com/identity/v3/services/5789da9864004dd088fce14c1c626a4b"
    },
    "enabled": true,
    "type": "volumev2",
    "id": "5789da9864004dd088fce14c1c626a4b",
    "description": "Block Storage Service V2"
  }
}
```

DELETE

/v3/services/{service_id}

Delete service

Relationship: <https://docs.openstack.org/api/openstack->

identity/3/rel/service

Deletes a service.

If you try to delete a service that still has associated endpoints, this call either deletes all associated endpoints or fails until all endpoints are deleted.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
service_id	path	string	The service ID.

GET

/v3/endpoints

List endpoints

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/endpoints>

Lists all available endpoints.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Request Parameters

Name	In	Type	Description
interface (Optional)	query	string	Filters the response by an interface.
service_id (Optional)	query	string	Filters the response by a service ID.

Response Parameters

Name	In	Type	Description
region_id	body	string	(Since v3.2) The ID of the region that contains the service endpoint.
links	body	object	The links for the <code>endpoints</code> resource.
url	body	string	The endpoint URL.
region	body	string	(Deprecated in v3.2) The geographic location of the service endpoint.
enabled	body	boolean	Indicates whether the endpoint appears in the service catalog: - <code>false</code> . The endpoint does not appear in the service catalog. - <code>true</code> . The endpoint appears in the service catalog.
interface	body	string	(Deprecated in v3.2) The geographic location of the service endpoint.
service_id	body	string	The UUID of the service to which the endpoint belongs.
endpoints	body	array	A list of <code>endpoint</code> objects.
id	body	string	The endpoint ID.

Response Example

```
{
  "endpoints": [
    {
      "enabled": true,
      "id": "0649c5be323f4792afbc1efdd480847d",
      "interface": "internal",
      "links": {
        "self": "http://example.com/identity/v3/endpoints/0649c5be323f4792afbc1efdd480847d"
      },
      "region": "RegionOne",
    }
  ]
}
```

```

    "region_id": "RegionOne",
    "service_id": "ef6b15e425814dc69d830361baae0e33",
    "url": "http://23.253.211.234:8080/v1/AUTH_$(tenant_id)s"
  },
  {
    "enabled": true,
    "id": "06b85ed2aa57413ca0b1813daed329a9",
    "interface": "internal",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/06b85ed2aa57413ca0b1813daed329a9"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "98cfd5347fb84601b2f88f3afd8dddd4",
    "url": "http://23.253.211.234:8776/v1/$(tenant_id)s"
  },
  {
    "enabled": true,
    "id": "070102f162e04f91a52c7887d0604163",
    "interface": "admin",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/070102f162e04f91a52c7887d0604163"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "312f401c14d143d8b3e3f4daf0418add",
    "url": "http://23.253.211.234:8774/v2.1/$(tenant_id)s"
  },
  {
    "enabled": true,
    "id": "0fd73b621e424cc0a172853264519cbc",
    "interface": "admin",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/0fd73b621e424cc0a172853264519cbc"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "17a877162c8e405b81d563d95ec4e3f8",
    "url": "http://23.253.211.234:8776/v2/$(tenant_id)s"
  },
  {
    "enabled": true,
    "id": "1899667a3b1544ccb355fdafc4184d7d7",
    "interface": "public",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/1899667a3b1544ccb355fdafc4184d7d7"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "9b67aed49e0d4c2fb46ca9476a3b9243",
    "url": "http://23.253.211.234:9292"
  },
  {
    "enabled": true,
    "id": "3b3611ea2e554ee7b85e7f2213b02c33",
    "interface": "admin",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/3b3611ea2e554ee7b85e7f2213b02c33"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "2a662f90700b4478929d4b24cc6a320b",
    "url": "http://23.253.211.234:9696/"
  },
  {
    "enabled": true,
    "id": "3ea2b420306f48c6bf0cf51c2fefea03",
    "interface": "internal",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/3ea2b420306f48c6bf0cf51c2fefea03"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "736fb9bb21ef498287db9abcc55b20d9",
    "url": "http://23.253.211.234:8774/v2/$(tenant_id)s"
  },
  {
    "enabled": true,
    "id": "41b122182f574a44b0e246aff6ca29c5",
    "interface": "admin",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/41b122182f574a44b0e246aff6ca29c5"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "9b67aed49e0d4c2fb46ca9476a3b9243",

```

```

    "url": "http://23.253.211.234:9292"
  },
  {
    "enabled": true,
    "id": "44a736dd5eeb4347acec66b5f11c8f80",
    "interface": "internal",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/44a736dd5eeb4347acec66b5f11c8f80"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "2a662f90700b4478929d4b24cc6a320b",
    "url": "http://23.253.211.234:9696/"
  },
  {
    "enabled": true,
    "id": "499e8f6718ef466ba3fb315fa8f9e0b8",
    "interface": "internal",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/499e8f6718ef466ba3fb315fa8f9e0b8"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "312f401c14d143d8b3e3f4daf0418add",
    "url": "http://23.253.211.234:8774/v2.1/$(tenant_id)s"
  },
  {
    "enabled": true,
    "id": "545b1e9f126248428c5cdbec7420c353",
    "interface": "public",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/545b1e9f126248428c5cdbec7420c353"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "736fb9bb21ef498287db9abcc55b20d9",
    "url": "http://23.253.211.234:8774/v2/$(tenant_id)s"
  },
  {
    "enabled": true,
    "id": "629dc5a64e954ad09a45e87bc48299ba",
    "interface": "public",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/629dc5a64e954ad09a45e87bc48299ba"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "2a662f90700b4478929d4b24cc6a320b",
    "url": "http://23.253.211.234:9696/"
  },
  {
    "enabled": true,
    "id": "642a329a660544fdaab2420c0da7d49b",
    "interface": "public",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/642a329a660544fdaab2420c0da7d49b"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "ef6b15e425814dc69d830361baae0e33",
    "url": "http://23.253.211.234:8080/v1/AUTH_$(tenant_id)s"
  },
  {
    "enabled": true,
    "id": "72f8fc8536e44a19bc3388218efcc741",
    "interface": "internal",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/72f8fc8536e44a19bc3388218efcc741"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "9b67aed49e0d4c2fb46ca9476a3b9243",
    "url": "http://23.253.211.234:9292"
  },
  {
    "enabled": true,
    "id": "74121e71962e4947ac622c41706f0ee7",
    "interface": "public",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/74121e71962e4947ac622c41706f0ee7"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "17a877162c8e405b81d563d95ec4e3f8",
    "url": "http://23.253.211.234:8776/v2/$(tenant_id)s"
  },

```

```

{
  "enabled": true,
  "id": "7431a4f971dc4abb8d0e387434a06817",
  "interface": "admin",
  "links": {
    "self": "http://example.com/identity/v3/endpoints/7431a4f971dc4abb8d0e387434a06817"
  },
  "region": "RegionOne",
  "region_id": "RegionOne",
  "service_id": "9242e05f0c23467bbd1cf1f7a6e5e596",
  "url": "http://23.253.211.234:8773/"
},
{
  "enabled": true,
  "id": "7cffc75a14ca4334b458e475750bd84f",
  "interface": "public",
  "links": {
    "self": "http://example.com/identity/v3/endpoints/7cffc75a14ca4334b458e475750bd84f"
  },
  "region": "RegionOne",
  "region_id": "RegionOne",
  "service_id": "efeb249cbcd3412496bc4b194ea058da",
  "url": "http://example.com/identity/v2.0"
},
{
  "enabled": true,
  "id": "a422a6fa163b4a6ba8309e067ce3750b",
  "interface": "public",
  "links": {
    "self": "http://example.com/identity/v3/endpoints/a422a6fa163b4a6ba8309e067ce3750b"
  },
  "region": "RegionOne",
  "region_id": "RegionOne",
  "service_id": "312f401c14d143d8b3e3f4daf0418add",
  "url": "http://23.253.211.234:8774/v2.1/$(tenant_id)s"
},
{
  "enabled": true,
  "id": "ac6a74efe9944afdb129d4df70cde0ec",
  "interface": "public",
  "links": {
    "self": "http://example.com/identity/v3/endpoints/ac6a74efe9944afdb129d4df70cde0ec"
  },
  "region": "RegionOne",
  "region_id": "RegionOne",
  "service_id": "9242e05f0c23467bbd1cf1f7a6e5e596",
  "url": "http://23.253.211.234:8773/"
},
{
  "enabled": true,
  "id": "adf43d7ff0d14d0fa1e8a5187f40e1af",
  "interface": "internal",
  "links": {
    "self": "http://example.com/identity/v3/endpoints/adf43d7ff0d14d0fa1e8a5187f40e1af"
  },
  "region": "RegionOne",
  "region_id": "RegionOne",
  "service_id": "efeb249cbcd3412496bc4b194ea058da",
  "url": "http://example.com/identity/v2.0"
},
{
  "enabled": true,
  "id": "b18be64a118244d39217db72534f8b33",
  "interface": "admin",
  "links": {
    "self": "http://example.com/identity/v3/endpoints/b18be64a118244d39217db72534f8b33"
  },
  "region": "RegionOne",
  "region_id": "RegionOne",
  "service_id": "736fb9bb21ef498287db9abcc55b20d9",
  "url": "http://23.253.211.234:8774/v2/$(tenant_id)s"
},
{
  "enabled": true,
  "id": "c828983c9c214d819674649aa693cdff",
  "interface": "public",
  "links": {
    "self": "http://example.com/identity/v3/endpoints/c828983c9c214d819674649aa693cdff"
  },
  "region": "RegionOne",
  "region_id": "RegionOne",
  "service_id": "98cfd5347fb84601b2f88f3afd8ddd4",
  "url": "http://23.253.211.234:8776/v1/$(tenant_id)s"
},
{
  "enabled": true,

```

```

      "id": "d062ebdb244f447498768fc0ced32e2d",
      "interface": "admin",
      "links": {
        "self": "http://example.com/identity/v3/endpoints/d062ebdb244f447498768fc0ced32e2d"
      },
      "region": "RegionOne",
      "region_id": "RegionOne",
      "service_id": "98cfd5347fb84601b2f88f3afd8dddd4",
      "url": "http://23.253.211.234:8776/v1/$(tenant_id)s"
    },
    {
      "enabled": true,
      "id": "d281219ec0df4cf2b7c681463d5dcf51",
      "interface": "internal",
      "links": {
        "self": "http://example.com/identity/v3/endpoints/d281219ec0df4cf2b7c681463d5dcf51"
      },
      "region": "RegionOne",
      "region_id": "RegionOne",
      "service_id": "17a877162c8e405b81d563d95ec4e3f8",
      "url": "http://23.253.211.234:8776/v2/$(tenant_id)s"
    },
    {
      "enabled": true,
      "id": "d8e0824a17404431b5d978a87ac1bede",
      "interface": "admin",
      "links": {
        "self": "http://example.com/identity/v3/endpoints/d8e0824a17404431b5d978a87ac1bede"
      },
      "region": "RegionOne",
      "region_id": "RegionOne",
      "service_id": "efeb249cbcd3412496bc4b194ea058da",
      "url": "http://example.com/identity_v2_admin/v2.0"
    },
    {
      "enabled": true,
      "id": "d9b54bdc063046828ac3c6487bea8047",
      "interface": "internal",
      "links": {
        "self": "http://example.com/identity/v3/endpoints/d9b54bdc063046828ac3c6487bea8047"
      },
      "region": "RegionOne",
      "region_id": "RegionOne",
      "service_id": "9242e05f0c23467bbd1cf1f7a6e5e596",
      "url": "http://23.253.211.234:8773/"
    },
    {
      "enabled": true,
      "id": "ea74f9771dec475eabfc2cdf5364413",
      "interface": "admin",
      "links": {
        "self": "http://example.com/identity/v3/endpoints/ea74f9771dec475eabfc2cdf5364413"
      },
      "region": "RegionOne",
      "region_id": "RegionOne",
      "service_id": "ef6b15e425814dc69d830361baae0e33",
      "url": "http://23.253.211.234:8080"
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "http://example.com/identity/v3/endpoints"
  }
}

```

POST

/v3/endpoints

Create endpoint

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/endpoints>

Creates an endpoint.

Response Codes

Success

Code	Reason
201 - Created	Resource was created and is ready to use.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
endpoint	body	object	An endpoint object.
url	body	string	The endpoint URL.
enabled (Optional)	body	boolean	Defines whether the endpoint appears in the service catalog: - <code>false</code> . The endpoint does not appear in the service catalog. - <code>true</code> . The endpoint appears in the service catalog. Default is <code>true</code> .
interface	body	string	The interface type, which describes the visibility of the endpoint. Value is: - <code>public</code> . Visible by end users on a publicly available network interface. - <code>internal</code> . Visible by end users on an unmetered internal network interface. - <code>admin</code> . Visible by administrative users on a secure network interface.
service_id	body	string	The UUID of the service to which the endpoint belongs.
region_id (Optional)	body	string	(Since v3.2) The ID of the region that contains the service endpoint.

Request Example

```
{
  "endpoint": {
    "interface": "public",
    "region_id": "RegionOne",
    "url": "http://example.com/identity/v3/endpoints/828384",
    "service_id": "9242e05f0c23467bbd1cf1f7a6e5e596"
  }
}
```

Response Parameters

Name	In	Type	Description
endpoint	body	object	An endpoint object.
links	body	object	The links for the endpoint resource.
url	body	string	The endpoint URL.
region	body	string	(Deprecated in v3.2) The geographic location of the service endpoint.
enabled	body	boolean	Indicates whether the endpoint appears in the service catalog: - <code>false</code> . The endpoint does not appear in the service catalog. - <code>true</code> . The endpoint appears in the service catalog.
interface	body	string	The interface type, which describes the visibility of the endpoint. Value is: - <code>public</code> . Visible by end users on a publicly available network interface. - <code>internal</code> . Visible by end users on an unmetered internal network interface. - <code>admin</code> . Visible by administrative users on a secure network interface.
service_id	body	string	The UUID of the service to which the endpoint belongs.
id	body	string	The endpoint ID.
region_id	body	string	(Since v3.2) The ID of the region that contains the service endpoint.

GET

/v3/endpoints/{endpoint_id}

Show endpoint details

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/endpoints>

Shows details for an endpoint.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
endpoint_id	path	string	The endpoint ID.

Response Parameters

Name	In	Type	Description
endpoint	body	object	An endpoint object.
name	body	string	The endpoint name.
links	body	object	The links for the endpoint resource.
url	body	string	The endpoint URL.
region	body	string	(Deprecated in v3.2) The geographic location of the service endpoint.
interface	body	string	The interface type, which describes the visibility of the endpoint. Value is: - public . Visible by end users on a publicly available network interface. - internal . Visible by end users on an unmetered internal network interface. - admin . Visible by administrative users on a secure network interface.
service_id	body	string	The UUID of the service to which the endpoint belongs.

Response Example

```
{
  "endpoint": {
    "enabled": true,
    "id": "01c3d5b92f7841ac83fb4b26173c12c7",
    "interface": "admin",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/01c3d5b92f7841ac83fb4b26173c12c7"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "3b2d6ad7e02c4cde8498a547601f1b8f",
    "url": "http://23.253.211.234:9696/"
  }
}
```

PATCH

/v3/endpoints/{endpoint_id}

Update endpoint

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/endpoint>

Updates an endpoint.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
endpoint	body	object	An endpoint object.
url	body	string	The endpoint URL.
region	body	string	(Deprecated in v3.2) The geographic location of the service endpoint.
interface	body	string	The interface type, which describes the visibility of the endpoint. Value is: - public . Visible by end users on a publicly available network interface. - internal . Visible by end users on an unmetered internal network interface. - admin . Visible by administrative users on a secure network interface.
service_id	body	string	The UUID of the service to which the

Name	In	Type	Description
			endpoint belongs.
endpoint_id	path	string	The endpoint ID.

Request Example

```
{
  "endpoint": {
    "interface": "public",
    "name": "Name",
    "region_id": "north",
    "url": "http://example.com/identity/v3/endpoints/828384",
    "service_id": "345678"
  }
}
```

Response Parameters

Name	In	Type	Description
endpoint	body	object	An endpoint object.
id	body	string	The endpoint ID.
links	body	object	The links for the <code>endpoint</code> resource.
url	body	string	The endpoint URL.
region	body	string	(Deprecated in v3.2) The geographic location of the service endpoint.
interface	body	string	The interface type, which describes the visibility of the endpoint. Value is: - <code>public</code> . Visible by end users on a publicly available network interface. - <code>internal</code> . Visible by end users on an unmetered internal network interface. - <code>admin</code> . Visible by administrative users on a secure network interface.
service_id	body	string	The UUID of the service to which the endpoint belongs.

Response Example

```
{
  "endpoint": {
    "id": "828384",
    "interface": "internal",
    "links": {
      "self": "http://example.com/identity/v3/endpoints/828384"
    },
    "region_id": "north",
    "service_id": "686766",
    "url": "http://example.com/identity/v3/endpoints/828384"
  }
}
```

DELETE

/v3/endpoints/{endpoint_id}

Delete endpoint

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/endpoint>

Deletes an endpoint.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
endpoint_id	path	string	The endpoint ID.

Users

A user is an individual API consumer that is owned by a domain. A role explicitly associates a user with projects or domains. A user with no assigned roles has no access to OpenStack resources.

You can list, create, show details for, update, delete, and change the password for users.

You can also list groups, projects, and role assignments for a specified user. To list user roles, see [Roles](#).

GET

/v3/users

List users

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/users>

Lists users.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.

Request Parameters

Name	In	Type	Description
domain_id (Optional)	query	string	Filters the response by a domain ID.
enabled (Optional)	query	string	Filters the response by either enabled (<code>true</code>) or disabled (<code>false</code>) users.
idp_id (Optional)	query	string	Filters the response by an identity provider ID.

Name	In	Type	Description
name (Optional)	query	string	Filters the response by a user name.
password_expires_at (Optional)	query	string	<p>Filter results based on which user passwords have expired. The query should include an operator and a timestamp with a colon (:) separating the two, for example:</p> <p>password_expires_at={operator}:{timestamp}</p> <ul style="list-style-type: none"> Valid operators are: lt, lte, gt, gte, eq, and neq <ul style="list-style-type: none"> lt: expiration time lower than the timestamp lte: expiration time lower than or equal to the timestamp gt: expiration time higher than the timestamp gte: expiration time higher than or equal to the timestamp eq: expiration time equal to the timestamp neq: expiration time not equal to the timestamp Valid timestamps are of the form: YYYY-MM-DDTHH:mm:ssZ. <p>For example:</p> <p>/v3/users?password_expires_at=lt:2016-12-08T22:02:00Z</p> <p>The example would return a list of users whose password expired before the timestamp (2016-12-08T22:02:00Z).</p>
protocol_id (Optional)	query	string	Filters the response by a protocol ID.
unique_id (Optional)	query	string	Filters the response by a unique ID.

Response Parameters

Name	In	Type	Description
links	body	object	The link to the collection of resources.
users	body	array	A list of user object, each containing:
default_project_id (Optional)	body	string	The ID of the default project for the user.
domain_id	body	string	The ID of the domain.
enabled	body	boolean	If the user is enabled, this value is true. If the user is disabled, this value is false.
id	body	string	The user ID.

Name	In	Type	Description
links	body	object	The links for the user resource.
name	body	string	The user name. Must be unique within the owning domain.
password_expires_at	body	string	<p>The date and time when the password expires. The time zone is UTC.</p> <p>This is a response object attribute; not valid for requests. A null value indicates that the password never expires.</p> <p>New in version 3.7</p>

Response Example

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://example.com/identity/v3/users"
  },
  "users": [
    {
      "domain_id": "default",
      "enabled": true,
      "id": "2844b2a08be147a08ef58317d6471f1f",
      "links": {
        "self": "http://example.com/identity/v3/users/2844b2a08be147a08ef58317d6471f1f"
      },
      "name": "glance",
      "password_expires_at": null
    },
    {
      "domain_id": "default",
      "enabled": true,
      "id": "4ab84ab39de54f4d96eaff8f2145a7cd",
      "links": {
        "self": "http://example.com/identity/v3/users/4ab84ab39de54f4d96eaff8f2145a7cd"
      },
      "name": "swiftusertest1",
      "password_expires_at": "2016-11-06T15:32:17.000000"
    },
    {
      "domain_id": "default",
      "enabled": true,
      "id": "56696a9a04864d63877a3d06a6f0b24b",
      "links": {
        "self": "http://example.com/identity/v3/users/56696a9a04864d63877a3d06a6f0b24b"
      },
      "name": "swift",
      "password_expires_at": null
    },
    {
      "domain_id": "default",
      "enabled": true,
      "id": "5acb638d15da44fc8de41b9a4bd41875",
      "links": {
        "self": "http://example.com/identity/v3/users/5acb638d15da44fc8de41b9a4bd41875"
      },
      "name": "alt_demo",
      "password_expires_at": "2016-11-06T15:32:17.000000"
    },
    {
      "domain_id": "default",
      "enabled": true,
      "id": "7596e862b1af473c8ed6ae99d35b51e3",
      "links": {
        "self": "http://example.com/identity/v3/users/7596e862b1af473c8ed6ae99d35b51e3"
      },
      "name": "demo",
      "password_expires_at": "2016-11-06T15:32:17.000000"
    }
  ],
}
```

```

{
  "domain_id": "default",
  "enabled": true,
  "id": "802edb2141b44e77bbde241417450749",
  "links": {
    "self": "http://example.com/identity/v3/users/802edb2141b44e77bbde241417450749"
  },
  "name": "nova",
  "password_expires_at": null
},
{
  "domain_id": "592ab0800d3745baaf45c610fa41950a",
  "enabled": true,
  "id": "9aca3883784647fe9aff3a50d922489a",
  "links": {
    "self": "http://example.com/identity/v3/users/9aca3883784647fe9aff3a50d922489a"
  },
  "name": "swiftusertest4",
  "password_expires_at": "2016-11-06T15:32:17.000000"
},
{
  "domain_id": "default",
  "enabled": true,
  "id": "a1251b011f9345e68c2458b841152034",
  "links": {
    "self": "http://example.com/identity/v3/users/a1251b011f9345e68c2458b841152034"
  },
  "name": "swiftusertest3",
  "password_expires_at": "2016-11-06T15:32:17.000000"
},
{
  "domain_id": "default",
  "enabled": true,
  "id": "a43f46eb318041f6b712143862e3ad70",
  "links": {
    "self": "http://example.com/identity/v3/users/a43f46eb318041f6b712143862e3ad70"
  },
  "name": "neutron",
  "password_expires_at": null
},
{
  "domain_id": "default",
  "enabled": true,
  "id": "b964a9e51c0046a4a84d3f83a135a97c",
  "links": {
    "self": "http://example.com/identity/v3/users/b964a9e51c0046a4a84d3f83a135a97c"
  },
  "name": "admin",
  "password_expires_at": null
},
{
  "domain_id": "default",
  "enabled": true,
  "id": "dc87e591c0d247d5ac04e873bd8a1646",
  "links": {
    "self": "http://example.com/identity/v3/users/dc87e591c0d247d5ac04e873bd8a1646"
  },
  "name": "cinder",
  "password_expires_at": null
},
{
  "domain_id": "default",
  "enabled": true,
  "id": "ed214dc1c2c6468b926c96eca6c8aee9",
  "links": {
    "self": "http://example.com/identity/v3/users/ed214dc1c2c6468b926c96eca6c8aee9"
  },
  "name": "glance-swift",
  "password_expires_at": "2016-11-06T15:32:17.000000"
},
{
  "domain_id": "default",
  "enabled": true,
  "id": "f4f6587b058a4f46a00242549b430d37",
  "links": {
    "self": "http://example.com/identity/v3/users/f4f6587b058a4f46a00242549b430d37"
  },
  "name": "swiftusertest2",
  "password_expires_at": "2016-11-06T15:32:17.000000"
}
]
}

```

POST
/v3/users

Create user

Relationship: <https://docs.openstack.org/api/openstack-identity/3/re1/users>

Creates a user.

Response Codes

Success

Code	Reason
201 - Created	Resource was created and is ready to use.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
user	body	object	A user object, containing:
default_project_id (Optional)	body	string	The ID of the default project for the user. Setting this attribute does not grant any actual authorization on the project, and is merely provided for convenience. Therefore, the referenced project does not need to exist within the user domain. (Since v3.1) If the user does not have authorization to their default project, the default project is ignored at token creation. (Since v3.1) Additionally, if your default project is not valid, a token is issued without an explicit scope of authorization.
domain_id (Optional)	body	string	The ID of the domain for the user.
enabled (Optional)	body	boolean	If the user is enabled, this value is <code>true</code> . If the user is disabled, this value is <code>false</code> .
name	body	string	The user name. Must be unique within the owning domain.
password (Optional)	body	string	The password for the user.

Request Example

```
{
  "user": {
    "default_project_id": "263fd9",
    "domain_id": "1789d1",
    "enabled": true,
    "name": "James Doe",
    "password": "secretsecret"
  }
}
```

Response Parameters

Name	In	Type	Description
user	body	object	A <code>user</code> object, containing:
default_project_id (Optional)	body	string	The ID of the default project for the user.
domain_id	body	string	The ID of the domain.
enabled	body	boolean	If the user is enabled, this value is <code>true</code> . If the user is disabled, this value is <code>false</code> .
id	body	string	The user ID.
links	body	object	The links for the <code>user</code> resource.
name	body	string	The user name. Must be unique within the owning domain.
password_expires_at	body	string	<p>The date and time when the password expires. The time zone is UTC.</p> <p>This is a response object attribute; not valid for requests. A <code>null</code> value indicates that the password never expires.</p> <p>New in version 3.7</p>

GET

`/v3/users/{user_id}`

Show user details

Relationship: <https://docs.openstack.org/api/openstack-identity/3/re1/user>

Shows details for a user.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
user_id	path	string	The user ID.

Response Parameters

Name	In	Type	Description
user	body	object	A <code>user</code> object, containing:
default_project_id (Optional)	body	string	The ID of the default project for the user.
domain_id	body	string	The ID of the domain.
enabled	body	boolean	If the user is enabled, this value is <code>true</code> . If the user is disabled, this value is <code>false</code> .
id	body	string	The user ID.
links	body	object	The links for the <code>user</code> resource.
name	body	string	The user name. Must be unique within the owning domain.
password_expires_at	body	string	<p>The date and time when the password expires. The time zone is UTC.</p> <p>This is a response object attribute; not valid for requests. A <code>null</code> value indicates that the password never expires.</p> <p>New in version 3.7</p>

Response Example

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

```

    "default_project_id": "263fd9",
    "domain_id": "1789d1",
    "enabled": true,
    "id": "9fe1d3",
    "links": {
        "self": "https://example.com/identity/v3/users/9fe1d3"
    },
    "name": "jsmith",
    "password_expires_at": "2016-11-06T15:32:17.000000"
}
}

```

PATCH

/v3/users/{user_id}

Update user

Relationship: <https://docs.openstack.org/api/openstack-identity/3/rel/user>

Updates a user's password, or whether they are enabled or disabled.

If the back-end driver does not support this functionality, this call might return the HTTP Not Implemented (501) response code.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.
501 - Not Implemented	The server either does not recognize the request method, or it lacks the ability to fulfill the request.

Request Parameters

Name	In	Type	Description
user_id	path	string	The user ID.

Name	In	Type	Description
user	body	object	A <code>user</code> object, containing:
default_project_id (Optional)	body	string	The new ID of the default project for the user.
domain_id (Optional)	body	string	The ID of the new domain for the user. The ability to change the domain of a user is now deprecated, and will be removed in subsequent release. It is already disabled by default in most Identity service implementations.
enabled (Optional)	body	boolean	Enables or disables the user. An enabled user can authenticate and receive authorization. A disabled user cannot authenticate or receive authorization. Additionally, all tokens that the user holds become no longer valid. If you reenable this user, pre-existing tokens do not become valid. To enable the user, set to <code>true</code> . To disable the user, set to <code>false</code> . Default is <code>true</code> .
name (Optional)	body	string	The new name for the user. Must be unique within the owning domain.
password (Optional)	body	string	The new password for the user.

Request Example

```
{
  "user": {
    "default_project_id": "263fd9",
    "enabled": true
  }
}
```

Response Parameters

Name	In	Type	Description
user	body	object	A <code>user</code> object, containing:
default_project_id (Optional)	body	string	The ID of the default project for the user.
domain_id	body	string	The ID of the domain.
enabled	body	boolean	If the user is enabled, this value is <code>true</code> . If the user is disabled, this value is <code>false</code> .
id	body	string	The user ID.
links	body	object	The links for the <code>user</code> resource.
name	body	string	The user name. Must be unique within the owning domain.

Name	In	Type	Description
password_expires_at	body	string	<p>The date and time when the password expires. The time zone is UTC.</p> <p>This is a response object attribute; not valid for requests. A <code>null</code> value indicates that the password never expires.</p> <p>New in version 3.7</p>

Response Example

```
{
  "user": {
    "default_project_id": "263fd9",
    "domain_id": "1789d1",
    "enabled": true,
    "id": "ff4e51",
    "links": {
      "self": "https://example.com/identity/v3/users/ff4e51"
    },
    "name": "jamesdoe",
    "password_expires_at": "2016-11-06T15:32:17.000000"
  }
}
```

DELETE

/v3/users/{user_id}

Delete user

Relationship: <https://docs.openstack.org/api/openstack-identity/3/re1/user>

Deletes a user.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
user_id	path	string	The user ID.

GET

/v3/users/{user_id}/groups

List groups to which a user belongs

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/user_groups

Lists groups to which a user belongs.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
user_id	path	string	The user ID.

Response Example

```
{
  "groups": [
    {
      "description": "Developers cleared for work on all general projects",
      "domain_id": "1789d1",
      "id": "ea167b",
      "links": {
        "self": "https://example.com/identity/v3/groups/ea167b"
      },
      "name": "Developers"
    },
    {
      "description": "Developers cleared for work on secret projects",
      "domain_id": "1789d1",
      "id": "a62db1",
      "links": {
```

```

        "self": "https://example.com/identity/v3/groups/a62db1"
      },
      "name": "Secure Developers"
    }
  ],
  "links": {
    "self": "http://example.com/identity/v3/users/9fe1d3/groups",
    "previous": null,
    "next": null
  }
}

```

GET

/v3/users/{user_id}/projects

List projects for user

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/user_projects

List projects for a user.

Response Codes

Success

Code	Reason
200 - OK	Request was successful.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.

Request Parameters

Name	In	Type	Description
user_id	path	string	The user ID.

Response Example

```

{
  "projects": [
    {
      "description": "description of this project",
      "domain_id": "161718",
      "enabled": true,
      "id": "456788",
      "links": {
        "self": "http://example.com/identity/v3/projects/456788"
      }
    }
  ]
}

```

```

    },
    "name": "a project name",
    "parent_id": "212223"
  },
  {
    "description": "description of this project",
    "domain_id": "161718",
    "enabled": true,
    "id": "456789",
    "links": {
      "self": "http://example.com/identity/v3/projects/456789"
    },
    "name": "another domain",
    "parent_id": "212223"
  }
],
"links": {
  "self": "http://example.com/identity/v3/users/313233/projects",
  "previous": null,
  "next": null
}
}

```

POST

/v3/users/{user_id}/password

Change password for user

Relationship: https://docs.openstack.org/api/openstack-identity/3/rel/user_change_password

Changes the password for a user.

Note

This API call does not require a token for authentication.

Response Codes

Success

Code	Reason
204 - No Content	The server has fulfilled the request by deleting the resource.

Error

Code	Reason
400 - Bad Request	Some content in the request was invalid.
401 - Unauthorized	User must authenticate before making a request.
403 - Forbidden	Policy does not allow current user to do this operation.
404 - Not Found	The requested resource could not be found.
409 - Conflict	This operation conflicted with another operation on this resource.

Request Parameters

Name	In	Type	Description
user_id	path	string	The user ID.
user	body	object	A <code>user</code> object, containing:
original_password	body	string	The original password for the user.
password	body	string	The new password for the user.

Request Example

```
{
  "user": {
    "password": "new_secretsecret",
    "original_password": "secretsecret"
  }
}
```