Elastic IP

API Reference(Ankara Region)

Issue	01
Date	2024-04-15

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Huawei Technologies Co., Ltd.

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

Website: https://www.huawei.com

Email: support@huawei.com

Security Declaration

Vulnerability

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

Contents

1 Before You Start

1.1 Overview

Welcome to *Elastic IP API Reference*. The EIP service provides independent public IP addresses and bandwidth for Internet access. EIPs can be bound to or unbound from ECSs, virtual IP addresses, NAT gateways, or load balancers. Various billing modes are provided to meet diverse service requirements.

This document describes how to use application programming interfaces (APIs) to perform operations on EIPs, such as creating, querying, deleting, and updating an EIP. For details about all supported operations, see **API Overview**.

If you plan to access EIPs through an API, ensure that you are familiar with EIP concepts. For details, see "Service Overview" in the *Elastic IP User Guide*.

1.2 API Calling

EIP supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see **Calling APIs**.

1.3 EIP Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. Contact the enterprise administrator for regions and endpoints.

1.4 Notes and Constraints

The number of EIPs that you can assign is determined by your quota. To view or increase the quota, see "What Is a Quota?" in *Elastic IP User Guide*.

1.5 Concepts

- Domain

  A domain has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The domain should not be used directly to perform routine management. For security purposes, create Identity and Access Management (IAM) users and grant them permissions for routine management.

- User

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

  API authentication requires information such as the domain name, username, and password.

- Region

  A region is a geographic area in which cloud resources are deployed. Availability zones (AZs) in the same region can communicate with each other over an intranet, while AZs in different regions are isolated from each other. Deploying cloud resources in different regions can better suit certain user requirements or comply with local laws or regulations.

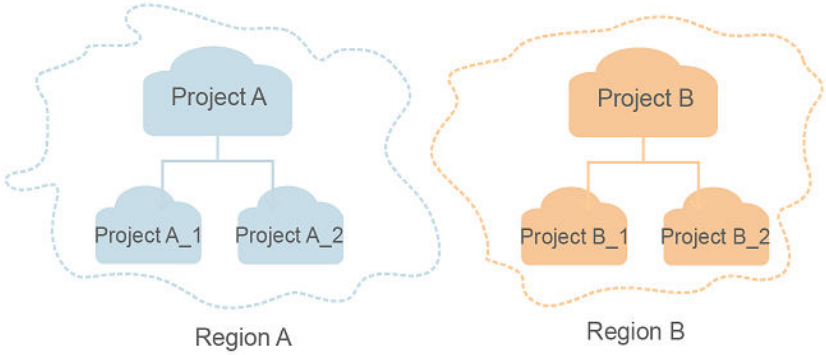
- AZ

  An AZ comprises of one or more physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Computing, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.

- Project

  A project corresponds to a region. Default projects are defined to group and physically isolate resources (including computing, storage, and network resources) across regions. Users can be granted permissions in a default project to access all resources under their domains in the region associated with the project. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

  Figure 1-1 Project isolation model



2 API Overview

APIs provided by the EIP service include native OpenStack APIs and EIP APIs.

A combination of these two types of APIs allows you to use all functions provided by the EIP service.

Table 2-1 EIP APIs

Type	Subtype	Description
EIP API	EIP	APIs for assigning, querying, updating, and releasing EIPs
EIP API	Floating IP address (IPv6)	APIs for assigning, querying, updating, and releasing IPv6 floating IP addresses
EIP API	Bandwidth	APIs for querying and updating bandwidth
EIP API	Bandwidth (v2.0)	• APIs for assigning and deleting shared bandwidth • APIs for adding an EIP to or removing an EIP from a shared bandwidth
EIP API	Quota	API for querying quota values
OpenStack Neutron API	Floating IP address	APIs for assigning, querying, updating, and releasing floating IP addresses

Type	Subtype	Description
OpenStack Neutron API	API version	APIs for querying all available API versions and displaying the results in pages.
EIP v3 API	EIP (v3)	APIs for binding and unbinding EIPs.

3 Calling APIs

3.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for obtaining a user token as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

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

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

Table 3-1 URI parameter description

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

NOTE

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

Request Methods

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

Table 3-2 HTTP methods

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

For example, in the case of the API used to obtain a user token, the request method is **POST**. The request is as follows:

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

Request Header

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

Common request header fields are as follows.

Table 3-3 Common request header fields

Parameter	Description	Mandatory	Example Value
Host	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of *Hostname:Port number*. If the port number is not specified, the default port is used. The default port number for **https** is **443**.	No This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443
Content-Type	Specifies the type (or format) of the message body. The default value **application/json** is recommended. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in **Obtaining a Project ID**.	No	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	Specifies the user token. It is a response to the API for obtaining a user token (This is the only API that does not require authentication). After the request is processed, the value of **X-Subject-Token** in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZIhvcNAQcCo...ggg1BBIINPXsidG9rZ

> **NOTE**
>
> In addition to supporting authentication using tokens, APIs support authentication using AK/SK, which uses SDKs to sign a request. During the signature, the **Authorization** (signature authentication) and **X-Sdk-Date** (time when a request is sent) headers are automatically added in the request.
>
> For more details, see "Authentication Using AK/SK" in **Authentication**.

The API used to obtain a user token does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

```
POST https://{{endpoint}}/v3/auth/tokens
Content-Type: application/json
```

(Optional) Request Body

This part is optional. The body of a request is often sent in a structured format (for example, JSON or XML) as specified in the **Content-Type** header field. The request body transfers content except the request header.

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

In the case of the API used to obtain a user token, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace *username*, *domainname*, *$ADMIN_PASS* (login password), and *xxxxxxxxxxxxxxxxx* (project name) with the actual values. Obtain a project name from the administrator.

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

```
POST https://{{endpoint}}/v3/auth/tokens
Content-Type: application/json

{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "$ADMIN_PASS",     //You are advised to store it in ciphertext in the configuration file or an environment variable and decrypt it when needed to ensure security.
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxxxxxxxxxxx"
            }
        }
```



```
    }
}
```

If all data required for the API request is available, you can send the request to call the API through **curl**, **Postman**, or coding. In the response to the API used to obtain a user token, **X-Subject-Token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

3.2 Authentication

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

- Token authentication: Requests are authenticated using tokens.
- AK/SK authentication: Requests are encrypted using AK/SK pairs. AK/SK authentication is recommended because it is more secure than token authentication.

Token Authentication

NOTE

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

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API. You can obtain a token by calling the Obtaining User Token API.

EIP is a project-level service. When you call the API, set **auth.scope** in the request body to **project**.

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",   // IAM user name
                    "password": "********",  // IAM user password
                    "domain": {
                        "name": "domainname"  //Name of the account to which the IAM user belongs
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxx"    // Project Name
            }
        }
    }
}
```

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

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



```
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK Authentication

An AK/SK is used to verify the identity of a request sender. In AK/SK authentication, a signature needs to be obtained and then added to requests.

NOTE

AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.

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

The following uses a demo project to show how to sign a request and use an HTTP client to send an HTTPS request.

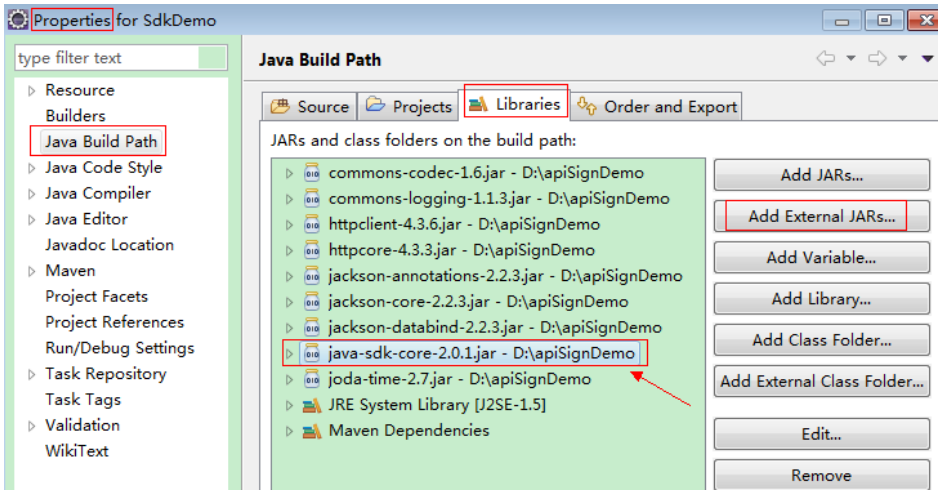
Download the demo project at **https://github.com/api-gate-way/SdkDemo**.

If you do not need the demo project, visit the following URL to download the API Gateway signing SDK:

Obtain the API Gateway signing SDK from the enterprise administrator.

Decompress the downloaded package to obtain a JAR file. Add the decompressed JAR file to the project as a dependency package. The following figure shows an example.

Figure 3-1 Introducing the API Gateway signing SDK



Step 1 Generate an AK/SK. (If you already have an AK/SK file, skip this step and find it. Generally, the file name is **credentials.csv**.)

1. Log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list.
3. In the navigation tree on the left, click **Access Keys**.
4. Click **Add Access Key**.

5. Enter an access key description and click **OK**.
6. Enter the verification code received by email, SMS message, or MFA application.

   NOTE

   If you have enabled operation protection (**Security Settings** > **Critical Operations** > **Operation Protection**), you need to enter the verification code.

   For users created in IAM that have not bound with any email address or mobile number, only the login password needs to be entered.
7. Download the access key file.

   NOTE

   Keep the access key secure.

Step 2 Download and extract the demo project.

Step 3 Import the demo project to Eclipse.

Figure 3-2 Selecting Existing Projects into Workspace

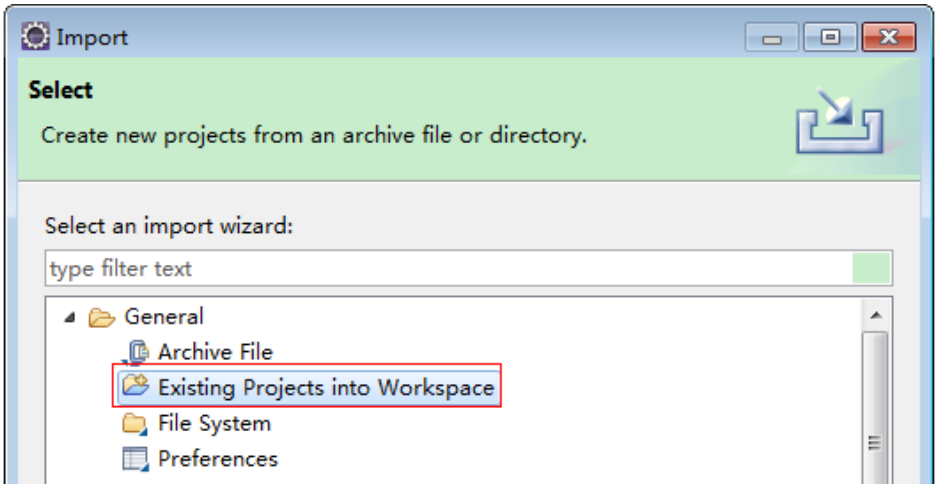


Figure 3-3 Selecting the demo project

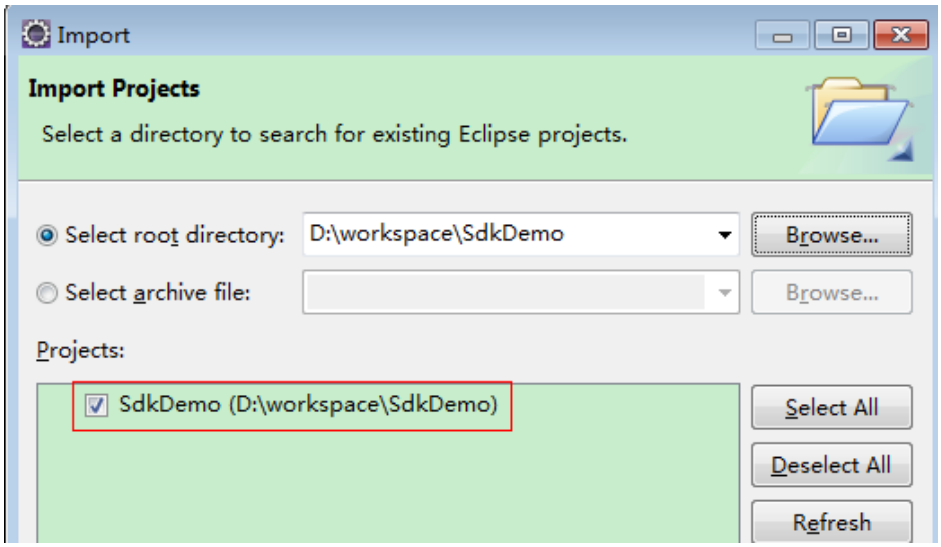
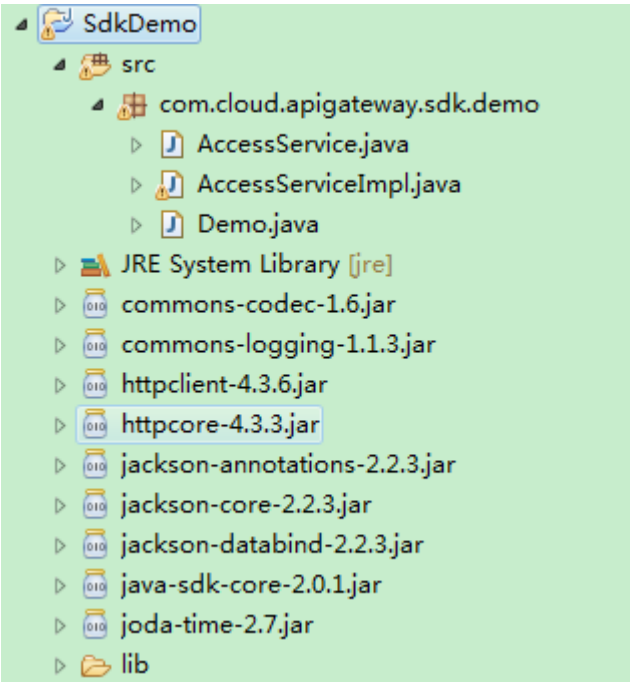


Figure 3-4 Structure of the demo project



Step 4 Sign the request.

The signature method is integrated into the JAR file imported in **Step 3**. Before sending the request, you need to sign the requested content. The obtained signature is included in the HTTP header of the request.

The demo code is classified into the following classes to demonstrate signing and sending the HTTP request:

- **AccessService**: An abstract class that merges the GET, POST, PUT, and DELETE methods into the **access** method.
- **Demo**: Execution entry used to simulate the sending of GET, POST, PUT, and DELETE requests.
- **AccessServiceImpl**: Implements the **access** method, which contains the code required for communication with API Gateway.

1. Edit the main method in the **Demo.java** file, and replace the bold text with actual values.

   If you use other methods such as POST, PUT, and DELETE, see the corresponding comment.

   Specify **region**, **serviceName**, **ak/sk**, and **url** as the actual values. In this demo, the URLs for accessing VPC resources are used.

   To obtain the project ID in the URLs, see **Obtaining a Project ID**.

   To obtain the endpoint, contact the enterprise administrator.

```
//TODO: Replace region with the name of the region in which the service to be accessed is located.
private static final String region = "";

//TODO: Replace vpc with the name of the service you want to access. For example, ecs, vpc, iam, and elb.
private static final String serviceName = "";

public static void main(String[] args) throws UnsupportedEncodingException
```

```
{
//TODO: Replace the AK and SK with those obtained on the My Credentials page.
String ak = "ZIRRKMTWP******1WKNKB";
String sk = "Us0mdMNHk******YrRCnW0ecfzl";

//TODO: To specify a project ID (multi-project scenarios), add the X-Project-Id header.
//TODO: To access a global service, such as IAM, DNS, CDN, and TMS, add the X-Domain-Id header to specify an account ID.
//TODO: To add a header, find "Add special headers" in the AccessServiceImple.java file.

//TODO: Test the API
String url = "https://{Endpoint}/v1/{project_id}/vpcs";
get(ak, sk, url);

//TODO: When creating a VPC, replace {project_id} in postUrl with the actual value.
//String postUrl = "https://serviceEndpoint/v1/{project_id}/cloudservers";
//String postbody ="{\"vpc\": {\"name\": \"vpc\",\"cidr\": \"192.168.0.0/16\"}}";
//post(ak, sk, postUrl, postbody);

//TODO: When querying a VPC, replace {project_id} in url with the actual value.
//String url = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//get(ak, sk, url);

//TODO: When updating a VPC, replace {project_id} and {vpc_id} in putUrl with the actual values.
//String putUrl = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//String putbody ="{\"vpc\":{\"name\": \"vpc1\",\"cidr\": \"192.168.0.0/16\"}}";
//put(ak, sk, putUrl, putbody);

//TODO: When deleting a VPC, replace {project_id} and {vpc_id} in deleteUrl with the actual values.
//String deleteUrl = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//delete(ak, sk, deleteUrl);
}
```

2. Compile the code and call the API.

   In the **Package Explorer** area on the left, right-click **Demo.java**, choose **Run AS** > **Java Application** from the shortcut menu to run the demo code.

   You can view API call logs on the console.

----End

3.3 Response

Status Code

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

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see **Status Codes**.

For example, if status code **201** is returned for calling the API used to obtain a user token, the request is successful.

Response Header

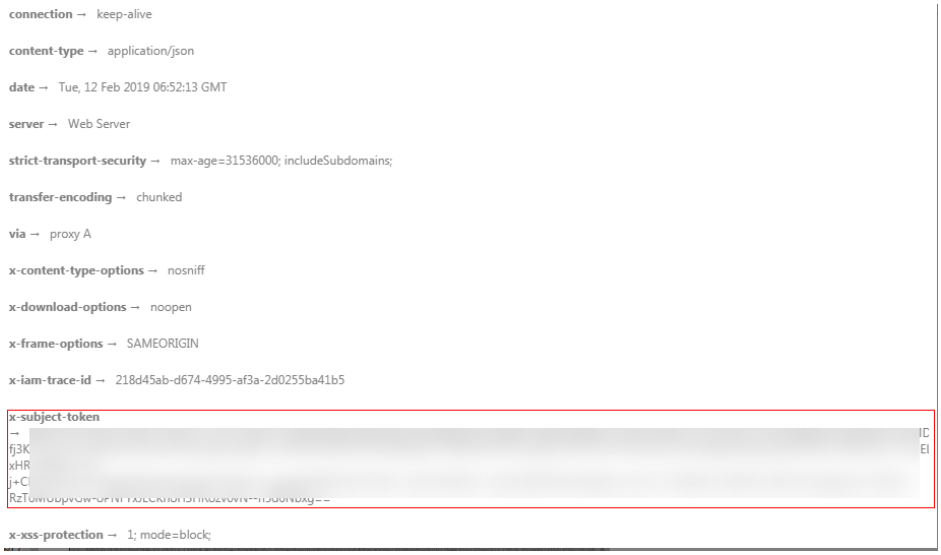
Similar to a request, a response also has a header, for example, **Content-Type**.

Figure 3-5 shows the response header fields for the API used to obtain a user token. The **X-Subject-Token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

NOTE

For security purposes, you are advised to set the token in ciphertext in configuration files or environment variables and decrypt it when using it.

Figure 3-5 Header fields of the response to the request for obtaining a user token



(Optional) Response Body

The body of a response is often returned in a structured format (for example, JSON or XML) as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to obtain a user token.

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

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

```
{
    "error_msg": "The request message format is invalid.",
    "error_code": "IMG.0001"
}
```

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

4 APIs

4.1 EIP

4.1.1 Assigning an EIP

Function

This API is used to assign an EIP.

The EIP service provides independent public IP addresses and bandwidth for Internet access. EIPs can be bound to or unbound from ECSs, virtual IP addresses, NAT gateways, or load balancers.

URI

POST /v1/{project_id}/publicips

Table 4-1 describes the parameters.

Table 4-1 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see **Obtaining a Project ID**.

Request Parameters

Table 4-2 Request body parameter

Name	Mandatory	Type	Description
publicip	Yes	**publicip** object	Specifies the EIP object. For details, see **Table 4-3**.
bandwidth	Yes	**bandwidth** object	Specifies the bandwidth object. For details, see **Table 4-4**.

Table 4-3 Description of the **publicip** field

Name	Mandatory	Type	Description
type	Yes	String	• Specifies the EIP type. • Constraints: – The configured value must be supported by the system. – **publicip_id** is an IPv4 port. If **publicip_type** is not specified, the default value is **5_bgp**.
alias	No	String	• Specifies the EIP name. • The value can contain 1 to 64 characters, including letters, digits, underscores (_), hyphens (-), and periods (.).

Table 4-4 Description of the **bandwidth** field

Name	Mandatory	Type	Description
name	Yes	String	• Specifies the bandwidth name. • The value can contain 1 to 64 characters, including letters, digits, underscores (_), hyphens (-), and periods (.). • This parameter is mandatory when **share_type** is set to **PER**. This parameter will be ignored when **share_type** is set to **WHOLE** with an ID specified.

Name	Mandatory	Type	Description
size	Yes	Integer	• Specifies the bandwidth size. • The value ranges from 1 Mbit/s to 300 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the bandwidth range of each region on the management console.) • This parameter is mandatory when **share_type** is set to **PER**. This parameter will be ignored when **share_type** is set to **WHOLE** with an ID specified. • The minimum increment for bandwidth adjustment varies depending on the bandwidth range. The details are as follows: – The minimum increment is 1 Mbit/s if the allowed bandwidth ranges from 0 Mbit/s to 300 Mbit/s (with 300 Mbit/s included). – The minimum increment is 50 Mbit/s if the allowed bandwidth ranges from 300 Mbit/s to 1000 Mbit/s (with

Name	Mandatory	Type	Description
			1000 Mbit/s included). – The minimum increment is 500 Mbit/s if the allowed bandwidth is greater than 1000 Mbit/s.
share_type	Yes	String	• Specifies the bandwidth type. • The value is **PER**, indicating that the bandwidth is dedicated.
charge_mode	No	String	• The value is **traffic**, indicating that the billing is based on traffic.

Example Request

Example request (IPv4 EIP with dedicated bandwidth)

```
POST https://{Endpoint}/v1/{project_id}/publicips

{
    "publicip": {
        "type": "5_bgp",
        "ip_version": 4
    },
    "bandwidth": {
        "name": "bandwidth123",
        "size": 10,
        "share_type": "PER"
    },
}
```

Response Message

- Response parameter

Table 4-5 Response parameter

Name	Type	Description
publicip	**publicip** object	Specifies the EIP object. For details, see **Table 4-6**.

Table 4-6 Description of the **publicip** field

Name	Type	Description
id	String	Specifies the unique identifier of an EIP.
status	String	• Specifies the EIP status. • Possible values are as follows: – **FREEZED** (Frozen) – **BIND_ERROR** (Binding failed) – **BINDING** (Binding) – **PENDING_DELETE** (Releasing) – **PENDING_CREATE** (Assigning) – **PENDING_UPDATE** (Updating) – **NOTIFYING** (Assigning) – **NOTIFY_DELETE** (Release) – **DOWN** (Unbound) – **ACTIVE** (Bound) – **ELB** (Bound to a load balancer) – **ERROR** (Exceptions)
type	String	• Specifies the EIP type. • Constraints: – The configured value must be supported by the system. – **publicip_id** is an IPv4 port. If **publicip_type** is not specified, the default value is **5_bgp**.
public_ip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.
tenant_id	String	Specifies the project ID.

Name	Type	Description
create_time	String	Specifies the time (UTC) when the EIP is assigned. Format: *yyyy-MM-dd HH:mm:ss*
bandwidth_size	Integer	Specifies the bandwidth (Mbit/s).
alias	String	Specifies the EIP name.

Example Response

Example response (IPv4 EIP with dedicated bandwidth)

```
{
   "publicip": {
      "id": "f588ccfa-8750-4d7c-bf5d-2ede24414706",
      "alias": "tom",
      "public_border_group": "center",
      "status": "PENDING_CREATE",
      "type": "5_bgp",
      "public_ip_address": "161.xx.xx.7",
      "tenant_id": "8b7e35ad379141fc9df3e178bd64f55c",
      "ip_version": 4,
      "create_time": "2015-07-16 04:10:52",
      "bandwidth_size": 0,

   }
}
```

Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

4.1.2 Querying an EIP

Function

This API is used to query a specific EIP.

URI

GET /v1/{project_id}/publicips/{publicip_id}

Table 4-7 describes the parameters.

Table 4-7 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see **Obtaining a Project ID**.
publicip_id	Yes	Specifies the unique identifier of an EIP.

Request Message

- Request parameter

  None

- Example request

```
Get https://{Endpoint}/v1/{project_id}/publicips/{publicip_id}
```

Response Message

- Response parameter

Table 4-8 Response parameter

Name	Type	Description
publicip	**publicip** object	Specifies the EIP object. For details, see **Table 4-9**.

Table 4-9 Description of the **publicip** field

Name	Type	Description
id	String	Specifies the unique identifier of an EIP.

Name	Type	Description
status	String	• Specifies the EIP status. • Possible values are as follows: – **FREEZED** (Frozen) – **BIND_ERROR** (Binding failed) – **BINDING** (Binding) – **PENDING_DELETE** (Releasing) – **PENDING_CREATE** (Assigning) – **PENDING_UPDATE** (Updating) – **NOTIFYING** (Assigning) – **NOTIFY_DELETE** (Releasing) – **DOWN** (Unbound) – **ACTIVE** (Bound) – **ELB** (Bound to a load balancer) – **ERROR** (Exceptions)
type	String	• Specifies the EIP type. • Constraints: – The configured value must be supported by the system. – **publicip_id** is an IPv4 port. If **publicip_type** is not specified, the default value is **5_bgp**.
private_ip_address	String	• Specifies the private IP address bound to the EIP. • This parameter is returned only if the private IP address is bound to the EIP.

Name	Type	Description
port_id	String	• Specifies the port ID. • This parameter is returned only when a port is associated with the EIP.
tenant_id	String	Specifies the project ID.
create_time	String	Specifies the time (UTC) when the EIP is assigned. Format: *yyyy-MM-dd HH:mm:ss*
bandwidth_id	String	Specifies the ID of the EIP bandwidth.
bandwidth_size	Integer	Specifies the bandwidth (Mbit/s).
bandwidth_share_type	String	• Specifies the EIP bandwidth type. • The value can be **PER** or **WHOLE**. – **PER**: Dedicated bandwidth – **WHOLE**: Shared bandwidth
alias	String	Specifies the EIP name.
public_border_group	String	Specifies whether it is in a central site or an edge site. The value can be: • center • *Edge site name* An EIP can only be bound to a resource of the same region.

- Example response

```
{
    "publicip": {
        "id": "2ec9b78d-9368-46f3-8f29-d1a95622a568",
        "status": "DOWN",
        "alias": "tom",
        "type": "5_bgp",
        "public_ip_address": "161.xx.xx.12",
        "tenant_id": "8b7e35ad379141fc9df3e178bd64f55c",
        "private_ip_address": "192.168.10.5",
        "create_time": "2015-07-16 04:32:50",
        "bandwidth_id": "49c8825b-bed9-46ff-9416-704b96d876a2",
```



```
        "bandwidth_share_type": "PER",
"bandwidth_size": 10,    //The EIP bandwidth size is 10 Mbit/s.
        "ip_version": 4
    }
}
```

Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

4.1.3 Querying EIPs

Function

This API is used to query EIPs.

URI

GET /v1/{project_id}/publicips

Table 4-10 describes the parameters.

Table 4-10 Parameter description

Name	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain a project ID, see **Obtaining a Project ID**.

Name	Mandatory	Type	Description
marker	No	String	Specifies a resource ID for pagination query, indicating that the query starts from the next record of the specified resource ID. This parameter can work together with the parameter **limit**. • If parameters **marker** and **limit** are not passed, resource records on the first page will be returned. • If the parameter **marker** is not passed and the value of parameter **limit** is set to **10**, the first 10 resource records will be returned. • If the value of the parameter **marker** is set to the resource ID of the 10th record and the value of parameter **limit** is set to **10**, the 11th to 20th resource records will be returned. • If the value of the parameter **marker** is set to the resource ID of the 10th record and the parameter **limit** is not passed, resource records starting from the 11th records (including 11th) will be returned.

Name	Mandatory	Type	Description
limit	No	Integer	Specifies the number of records that will be returned on each page. The value is from 0 to intmax (2^31-1). The default value is 2000. **limit** can be used together with **marker**. For details, see the parameter description of **marker**.

Request Message

- Request parameter

  None

- Example request

```
GET https://{Endpoint}/v1/{project_id}/publicips?limit={limit}&marker={marker}
```

Response Message

- Response parameter

Table 4-11 Response parameter

Name	Type	Description
publicips	Array of **publicips** objects	Specifies the EIP object. For details, see **Table 4-12**.

Table 4-12 Description of the **publicips** field

Name	Type	Description
id	String	Specifies the unique identifier of an EIP.

Name	Type	Description
status	String	• Specifies the EIP status. • Possible values are as follows: – **FREEZED** (Frozen) – **BIND_ERROR** (Binding failed) – **BINDING** (Binding) – **PENDING_DELETE** (Releasing) – **PENDING_CREATE** (Assigning) – **PENDING_UPDATE** (Updating) – **DOWN** (Unbound) – **ACTIVE** (Bound) – **ELB** (Bound to a load balancer) – **ERROR** (Exceptions)
type	String	• Specifies the EIP type. • Constraints: – The configured value must be supported by the system. – **publicip_id** is an IPv4 port. If **publicip_type** is not specified, the default value is **5_bgp**.
public_ip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.
private_ip_address	String	• Specifies the private IP address bound to the EIP. • This parameter is returned only if the private IP address is bound to the EIP.

Name	Type	Description
port_id	String	• Specifies the port ID. • This parameter is returned only when a port is associated with the EIP.
tenant_id	String	Specifies the project ID.
create_time	String	Specifies the time (UTC) when the EIP is assigned. Format: *yyyy-MM-dd HH:mm:ss*
bandwidth_id	String	Specifies the ID of the EIP bandwidth.
bandwidth_size	Integer	Specifies the bandwidth (Mbit/s).
bandwidth_share_type	String	• Specifies the EIP bandwidth type. • The value can be **PER** or **WHOLE**. – **PER**: Dedicated bandwidth – **WHOLE**: Shared bandwidth
alias	String	Specifies the EIP name.
public_border_group	String	Specifies whether it is in a central site or an edge site. The value can be: • center • *Edge site name* An EIP can only be bound to a resource of the same region.

- Example response

```
{
   "publicips": [
      {
         "id": "6285e7be-fd9f-497c-bc2d-dd0bdea6efe0",
         "status": "DOWN",
         "alias": "tom",
         "type": "5_bgp",
         "public_ip_address": "161.xx.xx.9",
         "private_ip_address": "192.168.10.5",
         "tenant_id": "8b7e35ad379141fc9df3e178bd64f55c",
         "create_time": "2015-07-16 04:22:32",
```



```
        "bandwidth_id": "3fa5b383-5a73-4dcb-a314-c6128546d855",
        "bandwidth_share_type": "PER",
        "bandwidth_size": 5,
        "ip_version": 4
    },
    {
        "id": "80d5b82e-43b9-4f82-809a-37bec5793bd4",
        "status": "DOWN",
        "type": "5_bgp",
        "public_ip_address": "161.xx.xx.10",
        "private_ip_address": "192.168.10.6",
        "tenant_id": "8b7e35ad379141fc9df3e178bd64f55c",
        "create_time": "2015-07-16 04:23:03",
        "bandwidth_id": "a79fd11a-047b-4f5b-8f12-99c178cc780a",
        "bandwidth_share_type": "PER",
        "bandwidth_size": 5,
        "ip_version": 4
    }
  ]
}
```


Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

4.1.4 Updating an EIP

Function

This API is used to convert the EIP version, bind an EIP to a NIC, or unbind an EIP from a NIC.

URI

PUT /v1/{project_id}/publicips/{publicip_id}

Table 4-13 describes the parameters.

Table 4-13 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see **Obtaining a Project ID**.
publicip_id	Yes	Specifies the unique identifier of an EIP.

Request Message

- Request parameter

Table 4-14 Request parameter

Name	Mandatory	Type	Description
publicip	Yes	**publicip** object	Specifies the EIP object. For details, see **Table 4-15**.

Table 4-15 Description of the **publicip** field

Name	Mandatory	Type	Description
port_id	No	String	• Specifies the port ID. • The value must be an existing port ID. If this parameter is not included or the parameter value is left blank, the EIP is unbound. If the specified port ID does not exist or has already been bound with an EIP, an error message will be displayed.
alias	No	String	• Specifies the EIP name. • The value can contain 1 to 64 characters, including letters, digits, underscores (_), hyphens (-), and periods (.).

- Example request 1 (Binding an EIP to a NIC)

```
PUT https://{Endpoint}/v1/{project_id}/publicips/{publicip_id}

{
    "publicip": {
        "port_id": "f588ccfa-8750-4d7c-bf5d-2ede24414706"
    }
}
```


Response Message

- Response parameter

Table 4-16 Response parameter

Name	Type	Description
publicip	**publicip** object	Specifies the EIP object. For details, see **Table 4-17**.

Table 4-17 Description of the **publicips** field

Name	Type	Description
id	String	Specifies the unique identifier of an EIP.
status	String	• Specifies the EIP status. • Possible values are as follows: – **FREEZED** (Frozen) – **BIND_ERROR** (Binding failed) – **BINDING** (Binding) – **PENDING_DELETE** (Releasing) – **PENDING_CREATE** (Assigning) – **PENDING_UPDATE** (Updating) – **NOTIFYING** (Assigning) – **NOTIFY_DELETE** (Releasing) – **DOWN** (Unbound) – **ACTIVE** (Bound) – **ELB** (Bound to a load balancer) – **ERROR** (Exceptions)

Name	Type	Description
type	String	• Specifies the EIP type. • Constraints: – The configured value must be supported by the system. – **publicip_id** is an IPv4 port. If **publicip_type** is not specified, the default value is **5_bgp**.
public_ip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.
private_ip_address	String	• Specifies the private IP address bound to the EIP. • This parameter is returned only when a port is associated with the EIP.
port_id	String	• Specifies the port ID. • This parameter is returned only when a port is associated with the EIP.
tenant_id	String	Specifies the project ID.
create_time	String	Specifies the time (UTC) when the EIP is assigned. Format: *yyyy-MM-dd HH:mm:ss*
bandwidth_id	String	Specifies the ID of the EIP bandwidth.
bandwidth_size	Integer	Specifies the bandwidth (Mbit/s).

Name	Type	Description
bandwidth_share_type	String	• Specifies the EIP bandwidth type. • The value can be **PER** or **WHOLE**. – **PER**: Dedicated bandwidth – **WHOLE**: Shared bandwidth
alias	String	Specifies the EIP name.

- Example response (Binding an EIP to a NIC)

```
{
  "publicip": {
    "id": "f6318bef-6508-4ea5-a48f-6152b6b1a8fb",
    "status": "ACTIVE",
    "alias": "tom",
    "type": "5_bgp",
    "port_id": "a135e9b8-1630-40d2-a6c5-eb534a61efbe",
    "public_ip_address": "10.xx.xx.162",
    "private_ip_address": "192.168.1.131",
    "tenant_id": "26ae5181a416420998eb2093aaed84d9",
    "create_time": "2019-03-27 01:33:18",
    "bandwidth_size": 7,
    "ip_version": 4,
    "bandwidth_name": "bandwidth-2aef",
    "bandwidth_share_type": "PER",
    "bandwidth_id": "7a258fff-10d8-44b8-8124-c59079eb8f4c"
  }
}
```

Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

4.1.5 Releasing an EIP

Function

This API is used to release an EIP.

URI

DELETE /v1/{project_id}/publicips/{publicip_id}

Table 4-18 describes the parameters.

Table 4-18 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see **Obtaining a Project ID**.
publicip_id	Yes	Specifies the unique identifier of an EIP.

Request Message

- Request parameter

  None

- Example request

  ```
  DELETE https://{Endpoint}/v1/{project_id}/publicips/{publicip_id}
  ```

Response Message

- Response parameter

  None

- Example response

  None

  Or

  ```
  {
      "code":"xxx",
      "message":"xxxxx"
  }
  ```

Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

4.2 Bandwidth

4.2.1 Querying a Bandwidth

Function

This API is used to query details about a bandwidth.

URI

GET /v1/{project_id}/bandwidths/{bandwidth_id}

Table 4-19 describes the parameters.

Table 4-19 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see **Obtaining a Project ID**.
bandwidth_id	Yes	Specifies the bandwidth ID, which uniquely identifies the bandwidth.

Request Message

- Request parameter

  None

- Example request

```
Get https://{Endpoint}//v1/{project_id}/bandwidths/{bandwidth_id}
```

Response Message

- Response parameter

Table 4-20 Response parameter

Name	Type	Description
bandwidth	**bandwidth** object	Specifies the bandwidth object.

Table 4-21 Description of the **bandwidth** field

Name	Type	Description
name	String	• Specifies the bandwidth name. • The value can contain 1 to 64 characters, including letters, digits, underscores (_), hyphens (-), and periods (.).

Name	Type	Description
size	Integer	• Specifies the bandwidth size. • The value ranges from 1 Mbit/s to 300 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the bandwidth range of each region on the management console.)
id	String	Specifies the bandwidth ID, which uniquely identifies the bandwidth.
share_type	String	• The value is **PER**, indicating that the bandwidth is dedicated.
publicip_info	Array of **publicip_info** objects	• Specifies information about the EIP that uses the bandwidth. For details, see **Table 4-22**.
tenant_id	String	Specifies the project ID.
bandwidth_type	String	• Specifies the bandwidth type. • The value is **bgp**.
charge_mode	String	• Specifies whether the billing is based on traffic or bandwidth. • Possible values can be **bandwidth** (billed by bandwidth) and **traffic** (billed by traffic). If the value is an empty character string or no value is specified, value **bandwidth** is used.
status	String	• Specifies the bandwidth status. • Possible values are as follows: – **FREEZED** (Frozen) – **NORMAL** (Normal)

Table 4-22 publicip_info object

Name	Type	Description
publicip_id	String	Specifies the ID of the EIP that uses the bandwidth.
publicip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.

Name	Type	Description
publicip_type	String	• Specifies the EIP type. • Constraints: – The configured value must be supported by the system. – **publicip_id** is an IPv4 port. If **publicip_type** is not specified, the default value is **5_bgp**.

- Example response

```
{
  "bandwidth": {
    "id": "3cbd5ae9-368f-4bc8-8841-f2ecc322c64a",
    "name": "EIPResourceSetup_1553594229",
    "size": 5,
    "share_type": "PER",
    "publicip_info": [
      {
        "publicip_id": "22b02f40-b95f-465a-ae9b-7c8b0f042a41",
        "publicip_address": "10.xx.xx.62",
        "ip_version": 4,
        "publicip_type": "5_bgp",
      }
    ],
    "tenant_id": "26ae5181a416420998eb2093aaed84d9",
    "bandwidth_type": "bgp",
    "charge_mode": "bandwidth",
    "status": "NORMAL",
    "created_at": "2020-04-21T07:58:02Z",
    "updated_at": "2020-04-21T07:58:02Z"
  }
}
```

Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

4.2.2 Querying Bandwidths

Function

This API is used to query bandwidths using search criteria.

URI

GET /v1/{project_id}/bandwidths

Table 4-23 describes the parameters.

Table 4-23 Parameter description

Name	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain a project ID, see **Obtaining a Project ID**.
marker	No	String	Specifies a resource ID for pagination query, indicating that the query starts from the next record of the specified resource ID. This parameter can work together with the parameter **limit**. • If parameters **marker** and **limit** are not passed, resource records on the first page will be returned. • If the parameter **marker** is not passed and the value of parameter **limit** is set to **10**, the first 10 resource records will be returned. • If the value of the parameter **marker** is set to the resource ID of the 10th record and the value of parameter **limit** is set to **10**, the 11th to 20th resource records will be returned. • If the value of the parameter **marker** is set to the resource ID of the 10th record and the parameter **limit** is not passed, resource records starting from the 11th records (including 11th) will be returned.
limit	No	Integer	Specifies the number of records that will be returned on each page. The value is from 0 to intmax (2^31-1). The default value is 2000. **limit** can be used together with **marker**. For details, see the parameter description of **marker**.

- Example request
  GET https://{Endpoint}/v1/{project_id}/bandwidths?limit={limit}&marker={marker}

Response Message

- Response parameter

Table 4-24 Response parameter

Name	Type	Description
bandwidths	Array of **bandwidths** objects	Specifies the bandwidth objects. For details, see **Table 4-25**.

Table 4-25 Description of the **bandwidths** field

Name	Type	Description
name	String	• Specifies the bandwidth name. • The value can contain 1 to 64 characters, including letters, digits, underscores (_), hyphens (-), and periods (.).
size	Integer	• Specifies the bandwidth size in Mbit/s. • The value ranges from 1 Mbit/s to 300 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the bandwidth range of each region on the management console.)
id	String	Specifies the bandwidth ID, which uniquely identifies the bandwidth.
share_type	String	• The value is **PER**, indicating that the bandwidth is dedicated. If this parameter is not set, the list of all bandwidths will be returned by default.
publicip_info	Array of **publicip_info** objects	• Specifies the information about the EIP that uses the bandwidth. For details, see **Table 4-26**.
tenant_id	String	Specifies the project ID.
bandwidth_type	String	• Specifies the bandwidth type. • The value is **bgp**.

Name	Type	Description
charge_mode	String	• Specifies whether the bandwidth is billed by traffic or by bandwidth size. • Possible values can be **bandwidth** (billed by bandwidth) and **traffic** (billed by traffic). If the value is an empty character string or no value is specified, value **bandwidth** is used.
status	String	• Specifies the bandwidth status. • Possible values are as follows: – **FREEZED** (Frozen) – **NORMAL** (Normal)

Table 4-26 publicip_info object

Name	Type	Description
publicip_id	String	Specifies the ID of the EIP that uses the bandwidth.
publicip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.
publicip_type	String	• Specifies the EIP type. • Constraints: – The configured value must be supported by the system. – **publicip_id** is an IPv4 port. If **publicip_type** is not specified, the default value is **5_bgp**.

- Example response

```
{
 "bandwidths": [
  {
   "id": "09b99c91-da7c-449f-94e2-f4934c5b2a71",
   "name": "test-f632a7b0-ef50-4ac5-97e9-ddc56b3d5977",
   "size": 200,
   "share_type": "PER",
   "publicip_info": [
    {
     "publicip_id": "2a65923c-7133-415d-ae3b-cf9635a942c5",
     "publicip_address": "10.xx.xx.3",
     "ip_version": 4,
     "publicip_type": "5_bgp"
    }
   ],
   "tenant_id": "26ae5181a416420998eb2093aaed84d9",
```



```
    "bandwidth_type": "bgp",
    "charge_mode": "bandwidth",
,
    "status": "NORMAL"
  },
  {
    "id": "0a583ff1-b43e-4000-ade3-e7af0097f832",
    "name": "test-7e880d5b-f458-40ad-a7e5-735c44cd8b7d",
    "size": 300,
    "share_type": "PER",
    "publicip_info": [
     {
       "publicip_id": "c754bc9a-16d5-4763-9674-d7561917aa80",
       "publicip_address": "10.xx.xx.9",
       "ip_version": 4,
       "publicip_type": "5_bgp"
     }
    ],
    "tenant_id": "26ae5181a416420998eb2093aaed84d9",
    "bandwidth_type": "bgp",
    "charge_mode": "bandwidth",
,
    "status": "NORMAL"
  },
  {
    "id": "0a673f00-3640-4a13-949e-7049b2916baf",
    "name": "bandwidth123",
    "size": 10,
    "share_type": "PER",
    "publicip_info": [
     {
       "publicip_id": "cec7fb70-2f82-4561-bd83-2121fb642fdc",
       "publicip_address": "10.xx.xx.184",
       "ip_version": 4,
       "publicip_type": "5_bgp"
     }
    ],
    "tenant_id": "26ae5181a416420998eb2093aaed84d9",
    "bandwidth_type": "bgp",
    "charge_mode": "bandwidth",
,
    "status": "NORMAL"
  },
  {
    "id": "0dde1eae-1783-46dc-998c-930fbe261ff9",
    "name": "bandwidth123",
    "size": 100,
    "share_type": "PER",
    "publicip_info": [
     {
       "publicip_id": "24232038-e178-40ad-80e4-5abb75db84be",
       "publicip_address": "10.xx.xx.101",
       "ip_version": 4,
       "publicip_type": "5_bgp"
     }
    ],
    "tenant_id": "26ae5181a416420998eb2093aaed84d9",
    "bandwidth_type": "bgp",
    "charge_mode": "bandwidth",
,
    "status": "NORMAL"
  }
 ]
}
```


Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

4.2.3 Updating a Bandwidth

Function

This API is used to update information about a bandwidth.

URI

PUT /v1/{project_id}/bandwidths/{bandwidth_id}

Table 4-27 describes the parameters.

Table 4-27 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see **Obtaining a Project ID**.
bandwidth_id	Yes	Specifies the bandwidth ID, which uniquely identifies the bandwidth.

Request Message

- Request parameter

Table 4-28 Request parameter

Name	Mandatory	Type	Description
bandwidth	Yes	**bandwidth** object	Specifies the bandwidth objects. For details, see **Table 4-29**.

Table 4-29 Description of the **bandwidth** field

Name	Mandatory	Type	Description
name	No	String	• Specifies the bandwidth name. • The value can contain 1 to 64 characters, including letters, digits, underscores (_), hyphens (-), and periods (.). If the value is left blank, the name of the bandwidth is not changed. • Either parameter **name** or **size** must be specified.
size	No	Integer	• Specifies the bandwidth size in Mbit/s. • The value ranges from 1 Mbit/s to 300 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the available bandwidth range on the management console.) If the parameter is not included, the bandwidth size is not changed. • Either parameter **name** or **size** must be specified. • If a decimal fraction (for example **10.2**) or a character string (for example **"10"**) is specified, the specified value will be automatically converted to an integer. • The minimum increment for bandwidth adjustment varies depending on the bandwidth range. The details are as follows: – The minimum increment is 1 Mbit/s if the allowed bandwidth ranges from 0 Mbit/s to 300 Mbit/s (with 300 Mbit/s included). – The minimum increment is 50 Mbit/s if the allowed bandwidth ranges from 300 Mbit/s to 1000 Mbit/s (with 1000 Mbit/s included). – The minimum increment is 500 Mbit/s if the allowed bandwidth is greater than 1000 Mbit/s.

- Example request


```
PUT https://{Endpoint}/v1/{project_id}/bandwidths/{bandwidth_id}

{
   "bandwidth":
      {"name": "bandwidth123",
       "size": 10
      }
}
```

Response Message

- Response parameter

Table 4-30 Response parameter

Name	Type	Description
bandwidth	**bandwidth** object	Specifies the bandwidth objects. For details, see **Table 4-31**.

Table 4-31 Description of the **bandwidth** field

Name	Type	Description
name	String	• Specifies the bandwidth name. • The value can contain 1 to 64 characters, including letters, digits, underscores (_), hyphens (-), and periods (.).
size	Integer	• Specifies the bandwidth size in Mbit/s. • The value ranges from 1Mbit/s~300Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the bandwidth range of each region on the management console.)
id	String	Specifies the bandwidth ID, which uniquely identifies the bandwidth.
share_type	String	• The value is **PER**, indicating that the bandwidth is dedicated.
publicip_info	Array of **publicip_info** objects	• Specifies the information about the EIP that uses the bandwidth. For details, see **Table 4-32**.
tenant_id	String	Specifies the project ID.
bandwidth_type	String	• Specifies the bandwidth type. • The value is **bgp**.

Name	Type	Description
charge_mode	String	• Specifies whether the bandwidth is billed by traffic or by bandwidth size. • Possible values can be **bandwidth** (billed by bandwidth) and **traffic** (billed by traffic). If the value is an empty character string or no value is specified, value **bandwidth** is used.

Table 4-32 publicip_info objects

Name	Type	Description
publicip_id	String	Specifies the ID of the EIP that uses the bandwidth.
publicip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.
publicip_type	String	• Specifies the EIP type. • Constraints: – The configured value must be supported by the system. – **publicip_id** is an IPv4 port. If **publicip_type** is not specified, the default value is **5_bgp**.

- Example response

```
{
   "bandwidth": {
      "id": "3fa5b383-5a73-4dcb-a314-c6128546d855",
      "name": "bandwidth123",
      "size": 10,
      "share_type": "PER",
      "publicip_info": [
         {
            "publicip_id": "6285e7be-fd9f-497c-bc2d-dd0bdea6efe0",
            "publicip_address": "161.xx.xx.9",
            "publicip_type": "5_bgp",
            "ip_version": 4
         }
      ],
      "tenant_id": "8b7e35ad379141fc9df3e178bd64f55c",
      "bandwidth_type": "bgp",
      "charge_mode": "bandwidth",
      "status": "NORMAL"
   }
}
```

Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

4.2.4 Updating Bandwidths in Batches

Function

This API is used to update bandwidths in batches. This API is not applicable to shared bandwidths and yearly/monthly bandwidths.

URI

PUT /v2/{project_id}/batch-bandwidths/modify

Table 4-33 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

Request Parameters

Table 4-34 Request body parameters

Parameter	Mandatory	Type	Description
bandwidths	Yes	Array of **ModifyBandwidthOption** objects	Update bandwidths.

Table 4-35 ModifyBandwidthOption

Parameter	Mandatory	Type	Description
id	Yes	String	• Bandwidth ID, which uniquely identifies a bandwidth Maximum length: **36**

Parameter	Mandatory	Type	Description
name	No	String	The name of the bandwidth. The value can contain 1 to 64 characters, including letters, digits, underscores (_), and hyphens (-). If the value is left blank, the name of the bandwidth is not changed. Either parameter **name** or **size** must be specified. Minimum length: **1** Maximum length: **64**
size	No	Integer	The bandwidth size, in Mbit/s. The value ranges from 1 Mbit/s to 2,000 Mbit/s by default. (The specific range may vary by the configuration in each region. You can see the available bandwidth range on the management console.) If the parameter is not specified, the bandwidth size is not changed. Either parameter **name** or **size** must be specified. If a decimal fraction (for example **10.2**) or a character string (for example **10**) is specified, the value will be automatically converted to an integer. The minimum increment for bandwidth adjustment varies with the bandwidth range. The minimum increment is 1 Mbit/s if the allowed bandwidth ranges from 0 Mbit/s to 300 Mbit/s. The minimum increment is 50 Mbit/s if the allowed bandwidth ranges 300 Mbit/s to 1,000 Mbit/s. The minimum increment is 500 Mbit/s if the allowed bandwidth is greater than 1,000 Mbit/s.

Parameter	Mandatory	Type	Description
charge_mode	No	String	Whether the bandwidth is billed by traffic, bandwidth, or 95th percentile bandwidth (enhanced). The value can be **bandwidth**, **traffic**, or **95peak_plus** (billed by enhanced 95th percentile bandwidth). If the value is an empty character string or no value is specified, value **bandwidth** is used. Only the shared bandwidth supports **95peak_plus** (billed by enhanced 95th percentile bandwidth). If you choose to be billed by 95th percentile bandwidth (enhanced), you need to specify the guaranteed bandwidth percentage. The default value is 20%. The value can be: • **bandwidth** • **traffic** • **95peak_plus**

Response Parameters

Status code: 200

Table 4-36 Response body parameters

Parameter	Type	Description
success_resources	Array of **SuccessResources** objects	Successful resources
failure_resources	Array of **FailureResources** objects	Failed resources

Table 4-37 SuccessResources

Parameter	Type	Description
id	String	• ID of the bandwidth that is successfully updated. Minimum length: **1** Maximum length: **36**

Table 4-38 FailureResources

Parameter	Type	Description
id	String	• ID of the bandwidth that fails to be updated. Minimum length: **1** Maximum length: **36**
code	String	• Error code Minimum length: **1** Maximum length: **36**
message	String	• Error message Minimum length: **1** Maximum length: **256**

Example Request

Update bandwidths in batches.

```
{
  "bandwidths" : [ {
    "id" : "837d84a0-b940-4401-9477-4a99de1979a7",
    "name" : "bandwidth123",
    "size" : 5
  }, {
    "id" : "f2549bed-c419-4f58-9609-7ade104772bb",
    "name" : "bandwidth123",
    "size" : 5
  } ]
}
```

Example Response

Status code: 200

Normal response to GET and PUT operations

```
{
  "success_resources" : [ {
    "id" : "837d84a0-b940-4401-9477-4a99de1979a7"
  } ],
  "failure_resources" : [ {
```



```
  "id" : "f2549bed-c419-4f58-9609-7ade104772bb",
  "code" : "VPC.0319",
  "message" : "updateBandwidth bandwidth failed. the bandwidth is share bandwidth."
 } ]
}
```

Status Codes

Status Code	Description
200	Normal response to GET and PUT operations

Error Codes

See **Error Codes**.

4.3 Quota

4.3.1 Querying the Quota

Function

This API is used to query the network resource quotas of a tenant. The network resources include VPCs, subnets, security groups, security group rules, EIPs, and more.

NOTE

This API can be used to query quotas of EIPs and VPCs.

URI

GET /v1/{project_id}/quotas

Example:

```
GET https://{Endpoint}/v1/{project_id}/quotas?type={type}
```

Table 4-39 describes the parameters.

Table 4-39 Parameter description

Name	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain a project ID, see **Obtaining a Project ID**.

Name	Mandatory	Type	Description
type	No	String	• Specifies the resource type. • Values: – **vpc**: VPC – **subnet**: Subnet – **securityGroup**: Security group – **securityGroupRule**: Security group rule – **publicIp**: EIP – **vpcPeer**: VPC peering connection – **loadbalancer**: Load balancer – **listener**: Load balancer listener – **physicalConnect**: Direct Connect connection – **virtualInterface**: Virtual interface – **firewall**: Firewall – **shareBandwidthIP**: IP address added to a shared bandwidth – **shareBandwidth**: Shared bandwidth – **address_group**: IP address group – **flow_log**: VPC fow log – **vpcContainRoutetable**: Number of route tables associated with a VPC – **routetableContainRoutes**: Number of routes in a route table

Request Parameters

None

Response Parameters

Table 4-40 Response parameter

Name	Type	Description
quotas	**quotas** object	Specifies the quota object. For details, see **Table 4-41**.

Table 4-41 Description of the **quotas** field

Name	Type	Description
resources	Array of **resource** objects	Specifies the resource objects. For details, see **Table 4-42**.

Table 4-42 Description of the **resource** field

Name	Type	Description
type	String	• Specifies the resource type. • Values: – **vpc**: VPC – **subnet**: Subnet – **securityGroup**: Security group – **securityGroupRule**: Security group rule – **publicIp**: EIP – **vpcPeer**: VPC peering connection – **loadbalancer**: Load balancer – **listener**: Load balancer listener – **physicalConnect**: Direct Connect connection – **virtualInterface**: Virtual interface – **firewall**: Firewall – **shareBandwidthIP**: IP address added to a shared bandwidth – **shareBandwidth**: Shared bandwidth – **address_group**: IP address group – **flow_log**: VPC fow log – **vpcContainRoutetable**: Number of route tables associated with a VPC – **routetableContainRoutes**: Number of routes in a route table

Name	Type	Description
used	Integer	• Specifies the number of created network resources. • The value ranges from **0** to the value of **quota**.
quota	Integer	• Specifies the maximum quota values for the resources. • The value ranges from the default quota value to the maximum quota value.
min	Integer	Specifies the minimum quota value allowed.

NOTE

If value **-1** is returned when you use an API to query your VPC quota, this indicates that the VPC quota is not limited.

Example Response


```
{
   "quotas": {
      "resources": [
         {
            "type": "vpc",
            "used": 4,
            "quota": 150,
            "min": 0
         },
         {
            "type": "subnet",
            "used": 5,
            "quota": 400,
            "min": 0
         },
         {
            "type": "securityGroup",
            "used": 1,
            "quota": 100,
            "min": 0
         },
         {
            "type": "securityGroupRule",
            "used": 6,
            "quota": 5000,
            "min": 0
         },
         {
            "type": "publicIp",
            "used": 2,
            "quota": 10,
            "min": 0
         },
         {
            "type": "vpcPeer",
            "used": 0,
            "quota": 50,
            "min": 0
         },
         {
            "type":"physicalConnect",
            "used":0,
```



```
            "quota":10,
            "min":0
        },
        {
            "type":"virtualInterface",
            "used":0,
            "quota":50,
            "min":0
        },
        {
            "type": "firewall",
            "used": 0,
            "quota": 200,
            "min": 0
        },
        {
            "type": "shareBandwidth",
            "used": 0,
            "quota": 5,
            "min": 0
        },
        {
            "type": "shareBandwidthIP",
            "used": 0,
            "quota": 20,
            "min": 0
        },
        {
            "type": "loadbalancer",
            "used": 0,
            "quota": 10,
            "min": 0
        },
        {
            "type": "listener",
            "used": 0,
            "quota": 10,
            "min": 0
        },
        {
            "type": "vpcContainRoutetable",
            "used": 0,
            "quota": 1,
            "min": 0
        },
        {
            "type": "routetableContainRoutes",
            "used": 0,
            "quota": 200,
            "min": 0
        },
        {
             "type": "address_group",
             "used": 0,
             "quota": 50,
             "min": 0
         }
    ]
  }
}
```


Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

5 Native OpenStack Neutron APIs V2.0

5.1 API Version Information

5.1.1 Querying API Versions

Function

This API is used to query all available versions of a native OpenStack API.

URI

GET /

Request Parameters

None

Example Request

```
GET https://{Endpoint}/
```

Response Parameters

Table 5-1 Response parameter

Parameter	Type	Description
versions	Array of **version** objects	Specifies the API version list. For details, see **Table 5-2**.

Table 5-2 version objects

Parameter	Type	Description
status	String	Specifies the API version status. Possible values are as follows: • **CURRENT** • **STABLE** • **DEPRECATED**
id	String	Specifies the API version.
links	Array of **link** objects	Specifies the link list. For details, see **Table 5-3**.

Table 5-3 link objects

Parameter	Type	Description
href	String	Specifies the API link.
rel	String	Specifies the relationship between the API link and the API version.

Example Response

```
{
   "versions": [
      {
         "status": "CURRENT",
         "id": "v2.0",
         "links": [
            {
               "href": "https://{Endpoint}/v2.0",
               "rel": "self"
            }
         ]
      }
   ]
}
```

Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

5.1.2 Pagination

Function

Neutron APIs v2.0 provides the pagination function. You can set parameters **limit** and **marker** in the URL to enable the desired number of items to be returned. All returned items are displayed in the ascending order of ID.

- To access the next page of the request, perform the following configurations:
  - Replace the value of **marker** in the original access request URL. Replace the value of **marker** to the value of **marker** in the value of **herf** if the value of **rel** in the response is **next**.
  - Set the value of **page_reverse** to **False**.
- To access the previous page of the request, perform the following configurations:
  - Replace the value of **marker** in the original access request URL. Replace the value of **marker** to the value of **marker** in the value of **herf** if the value of **rel** in the response is **previous**.
  - Set the value of **page_reverse** to **True**.

Request Parameters

Table 5-4 Request parameter

Parameter	Type	Mandatory	Description
limit	Integer	No	Specifies the number of items displayed per page.
marker	String	No	Specifies the ID of the last item in the previous list. If the marker value is invalid, error code 400 will be returned.
page_reverse	Boolean	No	Specifies the page direction. The value can be **True** or **False**.

Example Request

- When **page_reverse** is set to **False**:

```
GET https://{Endpoint}/v2.0/networks?limit=2&marker=3d42a0d4-a980-4613-ae76-a2cddecff054&page_reverse=False
```

- When **page_reverse** is set to **True**:

```
GET https://{Endpoint}/v2.0/vpc/peerings?limit=2&marker=e5a0c88e-228e-4e62-a8b0-90825b1b7958&page_reverse=True
```


Response Parameters

Table 5-5 Response parameter

Parameter	Type	Description
{resources}_links	Array of **{resources}_link** objects	Specifies the pagination information. For details, see **Table 5-6**. **{*resources*}** indicates the resource name, for example, **ports**, **networks**, **subnets**, **routers**, **firewall_rules**, **firewall_policies**, **firewall_groups**, **security_groups**, and **security_group_rules**. Only when **limit** is used for filtering and the number of resources exceeds the value of **limit** or 2000 (default value of **limit**), value **next** will be returned for **rel** and a link for **href**.

Table 5-6 {resources}_link object

Parameter	Type	Description
href	String	Specifies the API link.
rel	String	The API link is used to query the next or previous page. **next**: The next page is queried. **previous**: The previous page is queried.

Example Response

- When **page_reverse** is set to **False**:

```
{
   "networks": [
      {
         "status": "ACTIVE",
         "subnets": [],
         "name": "liudongtest ",
         "admin_state_up": false,
         "tenant_id": "6fbe9263116a4b68818cf1edce16bc4f",
         "id": "60c809cb-6731-45d0-ace8-3bf5626421a9"
      },
      {
         "status": "ACTIVE",
         "subnets": [
            "132dc12d-c02a-4c90-9cd5-c31669aace04"
         ],
         "name": "publicnet",
         "admin_state_up": true,
         "tenant_id": "6fbe9263116a4b68818cf1edce16bc4f",
         "id": "9daeac7c-a98f-430f-8e38-67f9c044e299"
      }
   ],
   "networks_links": [
```



```
    {
        "href": "http://192.168.82.231:9696/v2.0/networks?limit=2&marker=9daeac7c-
a98f-430f-8e38-67f9c044e299",
        "rel": "next"
    },
    {
        "href": "http://192.168.82.231:9696/v2.0/networks?limit=2&marker=60c809cb-6731-45d0-
ace8-3bf5626421a9&page_reverse=True",
        "rel": "previous"
    }
  ]
}
```


- When **page_reverse** is set to **True**:


```
{
  "peerings_links": [
    {
      "marker": "dd442819-5638-401c-bd48-a82703cf0464",
      "rel": "next"
    },
    {
      "marker": "1e13cbaf-3ce4-413d-941f-66d855dbfa7f",
      "rel": "previous"
    }
  ],
  "peerings": [
    {
      "status": "ACTIVE",
      "accept_vpc_info": {
        "vpc_id": "83a48834-b9bc-4f70-aa46-074568594650",
        "tenant_id": "e41a43bf06e249678413c6d61536eff9"
      },
      "request_vpc_info": {
        "vpc_id": "db8e7687-e43b-4fc1-94cf-16f69f484d6d",
        "tenant_id": "e41a43bf06e249678413c6d61536eff9"
      },
      "name": "peering1",
      "id": "1e13cbaf-3ce4-413d-941f-66d855dbfa7f"
    },
    {
      "status": "ACTIVE",
      "accept_vpc_info": {
        "vpc_id": "83a48834-b9bc-4f70-aa46-074568594650",
        "tenant_id": "e41a43bf06e249678413c6d61536eff9"
      },
      "request_vpc_info": {
        "vpc_id": "bd63cc9e-e7b8-4d4e-a0e9-055031470ffc",
        "tenant_id": "e41a43bf06e249678413c6d61536eff9"
      },
      "name": "peering2",
      "id": "dd442819-5638-401c-bd48-a82703cf0464"
    }
  ]
}
```


Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

5.2 Floating IP Address

5.2.1 Querying Floating IP Addresses

Function

This API is used to query all floating IP addresses accessible to the tenant submitting the request.

You can query the detailed information about a specified floating IP address using the API for **Querying a Floating IP Address**.

URI

GET /v2.0/floatingips

Table 5-7 describes the parameters.

Table 5-7 Parameter description

Parameter	Mandatory	Type	Description
id	No	String	Specifies the floating IP address ID.
floating_ip_address	No	String	Specifies the floating IPv4 address.
floating_network_id	No	String	Specifies the external network ID. You can only use fixed external network. You can use **GET /v2.0/networks?router:external=True** or **GET /v2.0/networks?name={floating_network}** or run the **neutron net-external-list** command to obtain information about the external network.
router_id	No	String	Specifies the ID of the belonged router.

Parameter	Mandatory	Type	Description
port_id	No	String	Specifies the port ID.
fixed_ip_address	No	String	Specifies the private IP address of the associated port.
tenant_id	No	String	Specifies the project ID.
limit	No	Integer	Specifies the number of records that will be returned on each page. The value is from 0 to intmax (2^31-1). The default value is 2000. **limit** can be used together with **marker**. For details, see the parameter description of **marker**.

Parameter	Mandatory	Type	Description
marker	No	String	Specifies a resource ID for pagination query, indicating that the query starts from the next record of the specified resource ID. This parameter can work together with the parameter **limit**. ● If parameters **marker** and **limit** are not passed, resource records on the first page will be returned. ● If the parameter **marker** is not passed and the value of parameter **limit** is set to **10**, the first 10 resource records will be returned. ● If the value of the parameter **marker** is set to the resource ID of the 10th record and the value of parameter **limit** is set to **10**, the 11th to 20th resource records will be returned. ● If the value of the parameter **marker** is set to the resource

Parameter	Mandatory	Type	Description
			ID of the 10th record and the parameter **limit** is not passed, resource records starting from the 11th records (including 11th) will be returned.
page_reverse	No	Boolean	Specifies the page direction. The value can be **True** or **False**.

Example:

```
GET https://{Endpoint}/v2.0/floatingips?
id={fip_id}&router_id={router_id}&floating_network_id={net_id}&floating_ip_address={floating_ip}&port_id={
port_id}&fixed_ip_address={fixed_ip}&tenant_id={tenant_id}
```

Request Message

None

Response Message

Table 5-8 Response parameter

Parameter	Type	Description
floatingips	Array of **floatingip** objects	Specifies the floating IP address list. For details, see **Table 5-9**.

Table 5-9 floatingip objects

Parameter	Type	Description
status	String	Specifies the floating IP address status. The value can be **ACTIVE**, **DOWN**, or **ERROR**. • **DOWN** indicates that the floating IP address has not been bound. • **ACTIVE** indicates that the floating IP address has been bound. • **ERROR** indicates that the floating IP address is abnormal.
id	String	Specifies the floating IP address ID.
project_id	String	Specifies the project ID.
floating_ip_address	String	Specifies the floating IP address.
floating_network_id	String	Specifies the external network ID.
router_id	String	Specifies the ID of the belonged router.
port_id	String	Specifies the port ID.
fixed_ip_address	String	Specifies the private IP address of the associated port.
tenant_id	String	Specifies the project ID.
created_at	String	Specifies the time when the floating IP address was created. UTC time is used. Format: *yyyy-MM-ddTHH:mm:ss*

Parameter	Type	Description
updated_at	String	Specifies the time when the floating IP address was updated. UTC time is used. Format: *yyyy-MM-ddTHH:mm:ss*

Example Request

```
GET https://{Endpoint}/v2.0/floatingips?limit=1
```

Example Response

Status code: 200

Normal response to the GET operation

```
{
  "floatingips" : [ {
    "id" : "1a3a2818-d9b4-4a9c-8a19-5252c499d1cd",
    "status" : "DOWN",
    "router_id" : null,
    "tenant_id" : "bbfe8c41dd034a07bebd592bf03b4b0c",
    "project_id" : "bbfe8c41dd034a07bebd592bf03b4b0c",
    "floating_network_id" : "0a2228f2-7f8a-45f1-8e09-9039e1d09975",
    "fixed_ip_address" : null,
    "floating_ip_address" : "99.99.99.84",
    "port_id" : null,


    "created_at" : "2017-10-19T12:21:28",
    "updated_at" : "2018-07-30T12:52:13"
  } ],
  "floatingips_links" : [ {
    "href" : "https://network.region.cn-test-2.clouds.com/v2.0/floatingips.json?
limit=2000&marker=000a6144-5010-46f2-bf06-6a1c94477ea3&page_reverse=true",
    "rel" : "previous"
  }, {
    "href" : "https://network.region.cn-test-2.clouds.com/v2.0/floatingips.json?limit=2000&marker=d445e537-
bc81-4039-9c7b-f9c1f5c73c78",
    "rel" : "next"
  } ]
}
```

Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

5.2.2 Querying a Floating IP Address

Function

This API is used to query details about a specified floating IP address, including the floating IP address status, ID of the router to which the floating IP address belongs, and external network ID of the floating IP address.

URI

GET /v2.0/floatingips/{floatingip_id}

Request Message

None

Response Message

Table 5-10 Response parameter

Parameter	Type	Description
floatingip	**floatingip** object	Specifies the floating IP address list. For details, see **Table 5-11**.

Table 5-11 floatingip objects

Attribute	Type	Description
status	String	Specifies the floating IP address status. The value can be **ACTIVE**, **DOWN**, or **ERROR**. • **DOWN** indicates that the floating IP address has not been bound. • **ACTIVE** indicates that the floating IP address has been bound. • **ERROR** indicates that the floating IP address is abnormal.

Attribute	Type	Description
id	String	Specifies the floating IP address ID.
project_id	String	Specifies the project ID.
floating_ip_address	String	Specifies the floating IP address.
floating_network_id	String	Specifies the external network ID.
router_id	String	Specifies the ID of the belonged router.
port_id	String	Specifies the port ID.
fixed_ip_address	String	Specifies the private IP address of the associated port.
tenant_id	String	Specifies the project ID.
created_at	String	Specifies the time when the floating IP address was created. UTC time is used. Format: *yyyy-MM-ddTHH:mm:ss*
updated_at	String	Specifies the time when the floating IP address was updated. UTC time is used. Format: *yyyy-MM-ddTHH:mm:ss*

Example Request

```
GET https://{Endpoint}/v2.0/floatingips/1a3a2818-d9b4-4a9c-8a19-5252c499d1cd
```

Example Response

Status code: 200

```
{
   "floatingip": {
      "id": "1a3a2818-d9b4-4a9c-8a19-5252c499d1cd",
      "status": "DOWN",
      "router_id": null,
      "tenant_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
      "project_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
      "floating_network_id": "0a2228f2-7f8a-45f1-8e09-9039e1d09975",
      "fixed_ip_address": null,
      "floating_ip_address": "99.99.99.84",
```



```
    "port_id": null,
    "created_at": "2017-10-19T12:21:28",
    "updated_at": "2018-07-30T12:52:13"
  }
}
```

Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

5.2.3 Assigning a Floating IP Address

Function

When assigning a floating IP address, you need to obtain the external network ID **floating_network_id** of the floating IP address.

You can use **GET /v2.0/networks?router:external=True** or run the **neutron net-external-list** command to obtain the UUID of the external network required for assigning a floating IP address.

URI

POST /v2.0/floatingips

Request Message

Table 5-12 Request parameter

Parameter	Type	Mandatory	Description
floatingip	**floatingip** object	Yes	Specifies the floating IP address list. For details, see **Table 5-13**.

Table 5-13 floatingip objects

Parameter	Mandatory	Type	Description
floating_ip_address	No	String	Specifies the floating IP address.

Parameter	Mandatory	Type	Description
floating_network_id	Yes	String	Specifies the external network ID. You can only use fixed external network. You can use **GET /v2.0/networks?router:external=True** or **GET /v2.0/networks?name={floating_network}** or run the **neutron net-external-list mode** command to obtain information about the external network.
port_id	No	String	Specifies the port ID.
fixed_ip_address	No	String	Specifies the private IP address of the associated port.

Response Message

Table 5-14 Response parameter

Parameter	Type	Description
floatingip	**floatingip** object	Specifies the floating IP address list. For details, see **Table 5-15**.

Table 5-15 floatingip objects

Attribute	Type	Description
status	String	Specifies the floating IP address status. The value can be **ACTIVE**, **DOWN**, or **ERROR**. • **DOWN** indicates that the floating IP address has not been bound. • **ACTIVE** indicates that the floating IP address has been bound. • **ERROR** indicates that the floating IP address is abnormal.

Attribute	Type	Description
id	String	Specifies the floating IP address ID.
floating_ip_address	String	Specifies the floating IP address.
floating_network_id	String	Specifies the external network ID.
router_id	String	Specifies the ID of the belonged router.
port_id	String	Specifies the port ID.
fixed_ip_address	String	Specifies the private IP address of the associated port.
tenant_id	String	Specifies the project ID.

Example Request

Create a floating IP address whose network is **0a2228f2-7f8a-45f1-8e09-9039e1d09975**.

```
POST https://{Endpoint}/v2.0/floatingips

{
   "floatingip": {
         "floating_network_id": "0a2228f2-7f8a-45f1-8e09-9039e1d09975"
   }
}
```

Example Response

Status code: 201

Normal response to POST requests

```
{
   "floatingip": {
      "id": "b997e0d4-3359-4c74-8f88-bc0af81cd5a2",
      "status": "DOWN",
      "router_id": null,
      "tenant_id": "bbfe8c41dd034a07bebd592bf03b4b0c",

      "floating_network_id": "0a2228f2-7f8a-45f1-8e09-9039e1d09975",
      "fixed_ip_address": null,
      "floating_ip_address": "88.88.215.205",
      "port_id": null,
   }
}
```

Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

5.2.4 Updating a Floating IP Address

Function

This API is used to update a floating IP address.

During the update, the ID of the floating IP address must be provided in the URL.

If **port_id** is left blank, the floating IP address has been unbound from the port.

NOTE

This API has the following constraints:

- If a floating IP address that you are binding is in the **error** state, unbind the IP address first.
- Do not associate a port that has a floating IP address associated to another floating IP address. You must first disassociate the port from the IP address and then associate it with another IP address.

URI

PUT /v2.0/floatingips/{floatingip_id}

Table 5-16 describes the parameters.

Table 5-16 Parameter description

Parameter	Mandatory	Type	Description
floatingip_id	Yes	String	Specifies the floating IP address ID. This parameter is not required when you assign a floating IP address. This parameter is mandatory when you query, update, or delete a floating IP address.

Request Message

Table 5-17 Request parameter

Parameter	Type	Mandatory	Description
floatingip	**floatingip** object	Yes	Specifies the floating IP address list. For details, see **Table 5-18**.

Table 5-18 floatingip objects

Parameter	Mandatory	Type	Description
port_id	No	String	Specifies the port ID.

Response Message

Table 5-19 Response parameter

Parameter	Type	Description
floatingip	**floatingip** object	Specifies the floating IP address list. For details, see **Table 5-20**.

Table 5-20 floatingip objects

Attribute	Type	Description
status	String	Specifies the floating IP address status. The value can be **ACTIVE**, **DOWN**, or **ERROR**. • **DOWN** indicates that the floating IP address has not been bound. • **ACTIVE** indicates that the floating IP address has been bound. • **ERROR** indicates that the floating IP address is abnormal.
id	String	Specifies the floating IP address ID.
floating_ip_address	String	Specifies the floating IP address.
floating_network_id	String	Specifies the external network ID.
router_id	String	Specifies the ID of the belonged router.
port_id	String	Specifies the port ID.

Attribute	Type	Description
fixed_ip_address	String	Specifies the private IP address of the associated port.
tenant_id	String	Specifies the project ID.

Example Request

- Unbind a floating IP address from a port.

```
PUT https://{Endpoint}/v2.0/floatingips/b997e0d4-3359-4c74-8f88-bc0af81cd5a2

{
   "floatingip": {
      "port_id": null
   }
}
```

- Bind a floating IP address to a port. The port ID is f91f5763-c5a2-4458-979d-61e48b3c3fac.

```
PUT https://{Endpoint}/v2.0/floatingips/b997e0d4-3359-4c74-8f88-bc0af81cd5a2

{
   "floatingip": {
         "port_id": "f91f5763-c5a2-4458-979d-61e48b3c3fac"
   }
}
```

Example Response

Status code: 200

(The floating IP address is unbound from the port.)

```
{
   "floatingip": {
      "id": "b997e0d4-3359-4c74-8f88-bc0af81cd5a2",
      "status": "DOWN",
      "router_id": null,
      "tenant_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
      "floating_network_id": "0a2228f2-7f8a-45f1-8e09-9039e1d09975",
      "fixed_ip_address": null,
      "floating_ip_address": "88.88.215.205",
      "port_id": null,
   }
}
```

(The floating IP address is bound to the port.)

```
{
   "floatingip": {
      "id": "b997e0d4-3359-4c74-8f88-bc0af81cd5a2",
      "status": "DOWN",
      "router_id": null,
      "tenant_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
      "floating_network_id": "0a2228f2-7f8a-45f1-8e09-9039e1d09975",
      "fixed_ip_address": "192.168.10.3",
      "floating_ip_address": "88.88.215.205",
      "port_id": "f91f5763-c5a2-4458-979d-61e48b3c3fac",
   }
}
```


Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

5.2.5 Deleting a Floating IP Address

Function

This API is used to delete a floating IP address.

URI

DELETE /v2.0/floatingips/{floatingip_id}

Table 5-21 describes the parameters.

Table 5-21 Parameter description

Parameter	Mandatory	Type	Description
floatingip_id	Yes	String	Specifies the floating IP address ID.

Request Message

None

Response Message

None

Example Request

Delete the floating IP address whose ID is a95ec431-8473-463b-aede-34fb048ee3a7.

```
DELETE https://{Endpoint}/v2.0/floatingips/a95ec431-8473-463b-aede-34fb048ee3a7
```

Example Response

None

Status Code

See **Status Codes**.

Error Code

See **Error Codes**.

6 Permissions Policies and Supported Actions

6.1 Introduction

You can use Identity and Access Management (IAM) for fine-grained permissions management of your EIP. If your does not need individual IAM users, you can skip this section.

By default, new IAM users do not have permissions assigned. You need to add them to one or more groups and attach policies or roles to these groups. The users then inherit permissions from the groups. This way, they can perform specified operations on cloud services based on the permissions.

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

Supported Actions

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

- Permissions: statements in a policy that allow or deny certain operations
- APIs: REST APIs that can be called by a user who has been granted specific permissions
- Actions: specific operations that are allowed or denied in a custom policy
- IAM projects/Enterprise projects: the authorization scope of a custom policy. A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions for both IAM and enterprise projects can be used and applied for both IAM and Enterprise Management. Policies that contain

actions only for IAM projects can be used and applied to IAM only. For details about the differences between IAM and enterprise projects, see .

NOTE

√: supported; x: not supported

EIP supports the following actions that can be defined in custom policies:

EIP actions that supported by EIP APIs include assigning an EIP, querying an EIP, querying EIPs, updating an EIP, and deleting an EIP.

6.2 EIP

Permission	API	Action	IAM Project
Assigns an EIP.	POST /v1/{project_id}/publicips	vpc:publicIps:create	√
Queries an EIP.	GET /v1/{project_id}/publicips/{publicip_id}	vpc:publicIps:get	√
Queries EIPs.	GET /v1/{project_id}/publicips	vpc:publicIps:list	√
Updates an EIP.	PUT /v1/{project_id}/publicips/{publicip_id}	vpc:publicIps:update	√
Release an EIP.	DELETE /v1/{project_id}/publicips/{publicip_id}	vpc:publicIps:delete	√

6.3 Bandwidth

Permission	API	Action	IAM Project
Queries a bandwidth.	GET /v1/{project_id}/bandwidths/{bandwidth_id}	vpc:bandwidths:get	√
Queries bandwidths.	GET /v1/{project_id}/bandwidths	vpc:bandwidths:list	√
Updates a bandwidth.	PUT /v1/{project_id}/bandwidths/{bandwidth_id}	vpc:bandwidths:update	√

6.4 Bandwidth (V2.0)

Permission	API	Action	IAM Project
Allocates a shared bandwidth.	POST /v2.0/{project_id}/bandwidths	vpc:bandwidths:create	√
Deletes a shared bandwidth.	DELETE /v2.0/{project_id}/bandwidths/{bandwidth_id}	vpc:bandwidths:delete	√
Adds an EIP to a shared bandwidth.	POST /v2.0/{project_id}/bandwidths/{bandwidth_id}/insert	vpc:publicIps:insert	√
Removes an EIP from a shared bandwidth.	POST /v2.0/{project_id}/bandwidths/{bandwidth_id}/remove	vpc:publicIps:remove	√

6.5 Floating IP Address (OpenStack Neutron API)

Permission	API	Action	IAM Project
Queries floating IP addresses.	GET /v2.0/floatingips	vpc:floatingIps:get	√
Queries a floating IP address.	GET /v2.0/floatingips/{floatingip_id}	vpc:floatingIps:get	√
Creates a floating IP address.	POST /v2.0/floatingips	vpc:floatingIps:create	√
Updates a floating IP address.	PUT /v2.0/floatingips/{floatingip_id}	vpc:floatingIps:update	√
Deletes a floating IP address.	DELETE /v2.0/floatingips/{floatingip_id}	vpc:floatingIps:delete	√

6.6 Precautions for API Permissions

If you have insufficient permissions, response code **200** will be returned when you query network resources and an empty list will be displayed.

A Appendix

A.1 VPC Monitoring Metrics

Description

This section describes monitoring metrics reported by VPC to Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the monitoring metrics of the monitored object and alarms generated for VPC.

Namespace

SYS.VPCnetwork ACL

Metrics

Table A-1 EIP and bandwidth metrics

ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
upstream_bandwidth	Outbound Bandwidth	Network rate of outbound traffic Unit: bit/s	≥ 0 bit/s	Bandwidth or EIP	1 minute
downstream_bandwidth	Inbound Bandwidth	Network rate of inbound traffic Unit: bit/s	≥ 0 bit/s	Bandwidth or EIP	1 minute

ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
up_stream	Outbound Traffic	Network traffic going out of the cloud platform in a minute Unit: byte	≥ 0 bytes	Bandwidth or EIP	1 minute
down_stream	Inbound Traffic	Network traffic going into the cloud platform in a minute Unit: byte	≥ 0 bytes	Bandwidth or EIP	1 minute

Dimension

Key	Value
publicip_id	EIP ID
bandwidth_id	Bandwidth ID

A.2 Status Codes

Table A-2 Normal values

Normal Response Code	Type	Description
200	OK	Specifies the normal response code for the GET, PUT, and POST operations.
201	Created	Specifies the normal response code for the POST operation of the OpenStack Neutron API and API V3.
204	No Content	Specifies the normal response code for the DELETE operation.

Table A-3 Abnormal values

Returned Value	Description
400 Bad Request	The server failed to process the request.
401 Unauthorized	You must enter a username and password to access the requested page.
403 Forbidden	You are forbidden to access the requested page.
404 Not Found	The server could not find the requested page.
405 Method Not Allowed	You are not allowed to use the method specified in the request.
406 Not Acceptable	The response generated by the server could not be accepted by the client.
407 Proxy Authentication Required	You must use the proxy server for authentication so that the request can be processed.
408 Request Timeout	The request timed out.
409 Conflict	The request could not be processed due to a conflict.
500 Internal Server Error	Failed to complete the request because of an internal service error.
501 Not Implemented	Failed to complete the request because the server does not support the requested function.
502 Bad Gateway	Failed to complete the request because the server has received an invalid response.
503 Service Unavailable	Failed to complete the request because the service is unavailable.
504 Gateway Timeout	A gateway timeout error occurred.

A.3 Error Codes

Description

If an error occurs when an API is called, error information is returned. This section describes the error information for EIP APIs (excluding native OpenStack APIs).

Example of Returned Error Information

```
{
    "code": "VPC.0504",
```



```
    "message": "Floating IP could not be found."
}
```

Error Code Description

Module	Status Code	Error Code	Message	Description	Handling Measure
Public	400	VPC.0002	Available zone Name is null.	The AZ is left blank.	Check whether the **availability_zone** field in the request body for creating a subnet is left blank.
	404	VPC.0003	VPC does not exist.	The VPC does not exist.	Check whether the VPC ID is correct or whether the VPC exists under the tenant.
	400	VPC.0004	VPC is not active, please try later.	The VPC status is abnormal.	Try again later or contact technical support.
	401	VPC.0005	Lack of user authority.	User restricted.	Check whether the account is in arrears or has not applied for the OBT permission.
	401	VPC.0009	real-name authentication fail.	Real-name authentication fails.	Contact technical support.
Public	400	VPC.0007	urlTenantId is not equal tokenTenantId	Inconsistent tenant IDs.	The tenant ID in the URL is different from that parsed in the token.

Module	Status Code	Error Code	Message	Description	Handling Measure
	401	VPC.0008	Invalid token in the header.	Invalid token.	Check whether the token in the request header is valid.
	403	VPC.2701	Token not allowed to do this action.	You do not have permission to perform this operation, or your account balance is insufficient.	Check whether the account balance is insufficient or whether your account has been frozen.
Creating a VPC	400	VPC.0101	Param is invalid.	VPC parameters are incorrect.	Check whether the parameter values are valid based on the returned error message and API reference document.
	409	VPC.0114	Quota exceeded for resources: ['router'].	The number of VPCs has reached the maximum allowed limit specified by the quota.	Clear VPC resources that no longer will be used or apply for expanding the VPC resource quota.
	400	VPC.0115	The router name has exist.	The VPC name already exists.	Change the VPC name.
Querying a VPC	400	VPC.0101	getVpc error vpcId is invalid.	VPC parameters are incorrect.	Ensure that the specified VPC ID is correct.

Module	Status Code	Error Code	Message	Description	Handling Measure
	404/500	VPC.0105	Neutron Error.	Calling the backend service fails.	Check whether the Neutron service is normal or contact technical support.
	500	VPC.0106	get router is null.	An error is returned for the failure to call the backend service.	Check whether the Neutron service is normal or contact technical support.
Querying VPCs	400	VPC.0101	Query vpc list error.	Failed to query the VPCs.	Check whether the parameter values are valid based on the returned error message.
	500	VPC.0105	Neutron Error.	Calling the backend service fails.	Check whether the Neutron service is normal or contact technical support.
	500	VPC.0106	query routers or getList are null.	The response result of calls to the IaaS OpenStack system is null or empty.	Check whether the Neutron service is normal or contact technical support.

Module	Status Code	Error Code	Message	Description	Handling Measure
Deleting a VPC	400/404	VPC.0101	Delete router error xx is invalid.	Invalid parameters.	Check whether the parameter values are valid based on the returned error message.
	500	VPC.0102	Delete router fail.	The interface fails to obtain the routing resources.	Contact technical support.
	409	VPC.0103	Resource status is busy. Try again later.	The VPC cannot be deleted because it is being created.	Contact technical support.
	409	VPC.0104	Router contains subnets, please delete subnet first.	The VPC cannot be deleted because it contains subnets.	Delete the subnet in the VPC.
	404/500	VPC.0105	Neutron Error.	Calling the backend service fails.	Check whether the Neutron service is normal or contact technical support.
	409	VPC.0107	Delete the firewall first before deleting the router.	Failed to delete the VPC because it has network ACLs associated.	Delete the network ACLs of the tenant first.

Module	Status Code	Error Code	Message	Description	Handling Measure
	409	VPC.0108	Router is used not allow deleted.	Failed to delete the VPC because it has EIPs associated.	Delete the EIPs of the tenant first.
	409	VPC.0110	deleteDefaultNetworkFromRouter router status is invalid.	The VPC cannot be deleted because its status is unstable.	Contact technical support.
	500	VPC.0111	Database Error.	An internal VPC exception occurs.	Contact technical support.
	409	VPC.0112	Delete the securitygroup first before deleting the router.	The VPC cannot be deleted because it contains security groups.	Delete security groups of the tenant.
	409	VPC.0118	ELB exists under this router, delete ELB firstly.	The VPC cannot be deleted because it contains load balancers.	Delete load balancers in the VPC.
	500	VPC.0119	ELB Error.	An error occurred when the VPC service makes calls to the ELB service.	Check whether the ELB service is normal or contact technical support.
	409	VPC.0120	exroutes exists under this router, delete exroutes firstly.	The VPC cannot be deleted because it contains extension routes.	Delete extension routes in the VPC.

Module	Status Code	Error Code	Message	Description	Handling Measure
Updating a VPC	400	VPC.0101	Update router xx is invalid.	Invalid parameters.	Check whether the parameter values are valid based on the returned error message.
	404/500	VPC.0105	Neutron Error.	Calling the backend service fails.	Check whether the Neutron service is normal or contact technical support.
	500	VPC.0113	Router status is not active.	The VPC cannot be updated because the status of the VPC is abnormal.	Try again later or contact technical support.
	400	VPC.0115	The router name has exist.	The VPC name already exists.	Change the VPC name.
	400	VPC.0117	Cidr can not contain subnetList cidr.	The subnet parameters are invalid. The VPC CIDR block does not contain all its subnet CIDR blocks.	Change the CIDR block of the VPC.

Module	Status Code	Error Code	Message	Description	Handling Measure
Creating a subnet	400	VPC.0201	Subnet name is invalid.	Incorrect subnet parameters.	Check whether the parameter values are valid based on the returned error message and API reference document.
	500	VPC.0202	Create subnet failed.	An internal error occurs in the subnet.	Contact technical support.
	400	VPC.0203	Subnet is not in the range of VPC.	The CIDR block of the subnet is not in the range of the VPC.	Change the CIDR block of the subnet.
	400	VPC.0204	The subnet has already existed in the VPC, or has been in conflict with the VPC subnet.	The CIDR block of the subnet already exists in the VPC.	Change the CIDR block of the subnet.
	400	VPC.0212	The subnet cidr is not valid.	Invalid subnet CIDR block.	Check whether the subnet CIDR block is valid.
Querying a subnet	400	VPC.0201	Subnet ID is invalid.	Invalid subnet ID.	Check whether the subnet ID is valid.
	404/500	VPC.0202	Query subnet fail.	Failed to query the subnet.	Contact technical support.

Module	Status Code	Error Code	Message	Description	Handling Measure
Querying subnets	400	VPC.0201	Query subnets list error.	Failed to query the subnets.	Check whether the parameter values are valid based on the returned error message.
	500	VPC.0202	List subnets error.	Failed to query the subnets.	Contact technical support.
Deleting a subnet	400	VPC.0201	Subnet ID is invalid.	Invalid subnet ID.	Check whether the parameter values are valid based on the returned error message.
	404/500	VPC.0202	Neutron Error.	An internal error occurs in the subnet.	Contact technical support.
	400	VPC.0207	Subnet does not belong to the VPC.	This operation is not allowed because the subnet does not belong to the VPC.	Check whether the subnet is in the VPC.
	500	VPC.0208	Subnet is used by private IP, can not be deleted.	The subnet cannot be deleted because it is being used by the private IP address.	Delete the private IP address of the subnet.

Module	Status Code	Error Code	Message	Description	Handling Measure
	500	VPC.0209	subnet is still used ,such as computer,LB.	The subnet cannot be deleted because it is being used by an ECS or load balancer.	Delete the ECS or load balancer in the subnet.
	500	VPC.0210	Subnet has been used by routes, please remove the routes first and try again.	The subnet cannot be deleted because it is being used by the custom route.	Delete the custom route.
	500	VPC.0211	subnet is still used by LBaas.	The subnet cannot be deleted because it is being used by load balancers.	Delete load balancers in the subnet.
Updating a subnet	400	VPC.0201	xx is invalid.	Incorrect subnet parameters.	Check whether the parameter values are valid based on the returned error message.
	404/500	VPC.0202	Neutron Error.	An internal error occurs in the subnet.	Contact technical support.
	500	VPC.0205	Subnet states is invalid, please try again later.	The subnet cannot be updated because it is being processed.	Try again later or contact technical support.

Module	Status Code	Error Code	Message	Description	Handling Measure
	400	VPC.0207	Subnet does not belong to the VPC.	This operation is not allowed because the subnet does not belong to the VPC.	Check whether the subnet is in the VPC.
Assigning an EIP	400	VPC.0301	Bandwidth name or share_type is invalid.	The specified bandwidth parameter for assigning an EIP is invalid.	Check whether the specified bandwidth parameter is valid.
	400	VPC.0501	Bandwidth share_type is invalid.	Invalid EIP parameters.	Check whether the parameter values are valid based on the returned error message and API reference document.
	403	VPC.0502	Tenant status is op_restricted.	You are not allowed to assign the EIP.	Check whether the account balance is insufficient or whether your account has been frozen.
	500	VPC.0503	Creating publicIp failed.	Failed to assign the EIP.	Contact technical support.
	500	VPC.0504	FloatIp is null.	Failed to assign the EIP because no IP address is found.	Contact technical support.

Module	Status Code	Error Code	Message	Description	Handling Measure
	500	VPC.0508	Port is invalid.	Port-related resources could not be found.	Contact technical support.
	409	VPC.0510	Floatingip has already associated with port.	The EIP has already been bound to another ECS.	Unbind the EIP from the ECS.
	409	VPC.0511	Port has already associated with floatingip.	The port has already been associated with an EIP.	Disassociate the port from the EIP.
	409	VPC.0521	Quota exceeded for resources: ['floatingip'].	Insufficient EIP quota.	Release the unbound EIPs or request to increase the EIP quota.
	409	VPC.0522	The IP address is in use.	The IP address is invalid or in use.	Check whether the IP address format is valid or replace it with another IP address.
	409	VPC.0532	No more IP addresses available on network.	Failed to assign the IP address because no IP addresses are available.	Release unbound EIPs or try again later.
Querying an EIP	400	VPC.0501	Invalid floatingip_id.	Invalid EIP parameters.	Check whether the EIP ID is valid.
	404	VPC.0504	Floating IP could not be found.	The EIP could not be found.	Check whether the specified EIP ID is valid.

Module	Status Code	Error Code	Message	Description	Handling Measure
	500	VPC.0514	Neutron Error.	An exception occurs in the IaaS OpenStack system.	Check whether the Neutron service is normal or contact technical support.
Querying EIPs	400	VPC.0501	Invalid limit.	Invalid EIP parameters.	Check whether the parameter values are valid based on the returned error message and API reference document.
Releasing an EIP	400	VPC.0501	Invalid param.	Invalid EIP parameters.	Contact technical support.
	404	VPC.0504	Floating IP could not be found.	The EIP could not be found.	Check whether the specified EIP ID is valid.
	409	VPC.0512	Resource status is busy, try it again later.	The EIP status is abnormal.	Try again later or contact technical support.
	500	VPC.0513	getElementByKey error.	Network resources cannot be found.	Contact technical support.
	500	VPC.0516	Publicip is in used by ELB.	Failed to release the EIP because it is being used by a load balancer.	Unbind the EIP from the load balancer.

Module	Status Code	Error Code	Message	Description	Handling Measure
	409	VPC.0517	Floatingip has associated with port, please disassociate it firstly.	Failed to release the EIP because it is bound to an ECS.	Unbind the EIP from the ECS.
	500	VPC.0518	Public IP has firewall rules.	Failed to release the EIP because it is being used by a network ACL.	Contact technical support.
Updating an EIP	400	VPC.0501	Port id is invalid.	Invalid EIP parameters.	Check whether the port ID is valid.
	404	VPC.0504	Floating IP could not be found.	The EIP could not be found.	Check whether the specified EIP ID is valid.
	500	VPC.0509	Floating ip double status is invalid.	The port has already been associated with an EIP.	Disassociate the port from the EIP.
	409	VPC.0510	Floatingip has already associated with port.	The EIP has already been bound to another ECS.	Unbind the EIP from the ECS.
	409	VPC.0511	Port has already associated with floatingip.	Failed to bind the EIP to the ECS because another EIP has already been bound to the ECS.	Unbind the EIP from the ECS.
	409	VPC.0512	Resource status is busy, try it again later.	The EIP status is abnormal.	Try again later or contact technical support.

Module	Status Code	Error Code	Message	Description	Handling Measure
	404/500	VPC.0514	Neutron Error.	An exception occurs in the IaaS OpenStack system.	Check whether the Neutron service is normal or contact technical support.
Querying a bandwidth	400	VPC.0301	getBandwidth error bandwidthId is invalid.	The bandwidth parameters are incorrect.	Check whether the bandwidth ID is valid.
	404	VPC.0306	No Eip bandwidth exist with id.	The bandwidth object does not exist.	The bandwidth object to be queried does not exist.
	500	VPC.0302	Neutron Error.	An exception occurs in the IaaS OpenStack system.	Check whether the Neutron service is normal or contact technical support.
Querying bandwidths	400	VPC.0301	Get bandwidths error limit is invalid.	The bandwidth parameters are incorrect.	Check whether the parameter values are valid based on the returned error message and API reference document.
	404	VPC.0306	No Eip bandwidth exist with id.	The bandwidth object does not exist.	The bandwidth object to be queried does not exist.

Module	Status Code	Error Code	Message	Description	Handling Measure
	500	VPC.0302	Neutron Error.	An exception occurs in the IaaS OpenStack system.	Check whether the Neutron service is normal or contact technical support.
Updating a bandwidth	400	VPC.0301	updateBandwidth input param is invalid.	The bandwidth parameters are incorrect.	Check whether the parameter values are valid based on the returned error message and API reference document.
	500	VPC.0302	Neutron Error.	Failed to obtain underlying resources.	Check whether the Neutron service is normal or contact technical support.
	500	VPC.0305	updateBandwidth error.	An internal error occurs during the bandwidth update.	Contact technical support.
Assigning a shared bandwidth	400	VPC.0310	NO QUOTAS for shareBandwidth!	Insufficient shared bandwidth quota.	Delete the shared bandwidth that is no longer required or contact technical support.

Module	Status Code	Error Code	Message	Description	Handling Measure
Adding an EIP to or removing an EIP from a shared bandwidth	400	VPC.0301	Invalid publicip_id	Invalid EIP.	Check whether the value of **publicip_id** in **publicip_info** is valid.
	400	VPC.0323	publicIp can not be operate with this bandwidth	Failed to add an EIP to or remove an EIP from a shared bandwidth.	Check whether the shared bandwidth or EIP is normal.
Querying quotas	400	VPC.1207	resource type is invalid.	The specified resource type does not exist.	Use an existing resource type.
Assigning a private IP address	500	VPC.0701	The IP has been used.	The private IP address already exists.	Change another private IP address and try again.
	400	VPC.0705	IP address is not a valid IP for the specified subnet.	Invalid private IP address	Check whether the specified IP address in the request body is within the subnet CIDR block.
	404	VPC.2204	Query resource by id fail.	The resource does not exist or the permission is insufficient.	Check whether the specified subnet in the request body exists or the current account has the permission to query the subnet.

Module	Status Code	Error Code	Message	Description	Handling Measure
	409	VPC.0703	No more IP addresses available on network xxx.	Insufficient IP addresses.	Check whether the subnet has sufficient IP addresses.
Querying a Private IP Address	404	VPC.0704	Query resource by id fail.	The private IP address does not exist.	Check whether the private IP address exists.
Querying Private IP Addresses	400	VPC.0702	query privateIps error.	Invalid parameters.	Check whether the parameter values are valid based on the returned error message.
Releasing a Private IP Address	404	VPC.0704	Query resource by id fail.	The private IP address does not exist.	Check whether the private IP address exists.
	500	VPC.0706	Delete port fail.	An error occurs when the private IP address is being released.	Try again later or contact technical support.
	409	VPC.0707	privateIp is in use.	The private IP address is in use.	Check whether the private IP address is being used by other resource.

Module	Status Code	Error Code	Message	Description	Handling Measure
Creating a security group	400	VPC.0601	Creating securitygroup name is invalid.	The parameters of the security group are incorrect.	Check whether the parameter values are valid based on the returned error message and API reference document.
	500	VPC.0602	Add security group fail.	An internal error occurs in the security group.	Check whether the Neutron service is normal or contact technical support.
	409	VPC.0604	Quota exceeded for resources: ['security_group'].	Insufficient security group quota.	Delete the security group that is no longer required or apply for increasing the quota.
Querying a security group	400	VPC.0601	Securitygroup id is invalid.	The parameters of the security group are incorrect.	Check whether the security group ID is valid.
	500	VPC.0602	Query security group fail.	An internal error occurs in the security group.	Check whether the Neutron service is normal or contact technical support.

Module	Status Code	Error Code	Message	Description	Handling Measure
	404	VPC.0603	Securitygroup is not exist.	The security group does not exist.	Check whether the security group ID is correct or whether the security group exists under the tenant.
	404/500	VPC.0612	Neutron Error.	An internal error occurs in the security group.	Contact technical support.
Querying security groups	400	VPC.0601	Query security groups error limit is invalid.	The parameters of the security group are incorrect.	Check whether the parameter values are valid based on the returned error message and API reference document.
	500	VPC.0602	Query security groups fail.	An internal error occurs in the security group.	Check whether the Neutron service is normal or contact technical support.

Module	Status Code	Error Code	Message	Description	Handling Measure
Creating a security group rule	409	VPC.0602	1.Security group rule already exists. 2.Quota exceeded for resources: ['security_group_rule']. 3.Failed to create the security group rule concurrently. The rule already exists.	The security group rule already exists. Insufficient security group rule quota. Failed to create the security group rule concurrentlly. The rule already exists.	Change the request body for creating a security group rule. Delete the security group rule that is no longer required or apply for increasing the quota. Check whether the security group rules created concurrently are different from each other.
Querying the network IP address usage	400	VPC.2301	parameter network_id is invalid.	The request parameter is incorrect.	Enter a valid network ID.
	400	VPC.2302	Network xxx could not be found.	The network is not found.	Ensure that the network ID exists.
Creating a VPC flow log	400	VPC.3001	resource_type/ log_store_type/ traffic_type/ log_group_id/ log_topic_id is invalid	Incorrect type or ID.	Check whether the type is supported or whether the ID format is correct.
	400	VPC.3002	Port does not support flow log, port id : xxx	The VPC flow log does not support this type of port.	Check whether the port is an S3, C3, or M3 ECS NIC port.

Module	Status Code	Error Code	Message	Description	Handling Measure
	404	VPC.3002	Port/Network/Vpc xxx could not be found.	The resource does not exist.	Check whether the resource exists.
	409	VPC.3004	Content of flow log is duplicate: resource type xxx, reousce id xxx, traffic type all, log group id xxx, log topic id xxx, log store type xxx, log store name xxx.	This VPC flow log already exists.	Modify the parameters of the VPC flow log.
	500	VPC.3002	Create flow log by xxx(tenant_id) fail.	Calling the backend service fails.	Try again later or contact technical support.
Querying VPC flow logs	404	VPC.3001	resource could not be found, xxx(listParam) is invalid	Invalid parameters.	Check whether the parameter format is correct.
	500	VPC.3002	Neutron Error.	Calling the backend service fails.	Try again later or contact technical support.
Querying a VPC flow log	404	VPC.3001	resource could not be found, flowlog id is invalid.	Invalid VPC flow log ID.	Check whether the VPC flow log ID format is correct.
	404	VPC.3002	Flow log xxx could not be found.	The VPC flow log does not exist.	Check whether the VPC flow log exists or whether its ID is correct.
Updating a VPC flow log	404	VPC.3001	resource could not be found, flowlog id is invalid.	Invalid VPC flow log ID.	Check whether the VPC flow log ID format is correct.

Module	Status Code	Error Code	Message	Description	Handling Measure
	404	VPC.3005	Flow log xxx could not be found.	The VPC flow log does not exist.	Check whether the VPC flow log exists or whether its ID is correct.
	500	VPC.3002	Update flow log by xxx(tenant_id) fail.	Calling the backend service fails.	Try again later or contact technical support.
Deleting a VPC flow log	404	VPC.3001	resource could not be found, flowlog id is invalid.	Invalid VPC flow log ID.	Check whether the VPC flow log ID format is correct.
	404	VPC.3005	Flow log xxx could not be found.	The VPC flow log does not exist.	Check whether the VPC flow log exists or whether its ID is correct.
	500	VPC.3002	Delete flow log by xxx(tenant_id) fail.	Calling the backend service fails.	Try again later or contact technical support.

A.4 Obtaining a Project ID

Scenarios

A project ID is required for some URLs when an API is called. Therefore, you need to obtain a project ID in advance. Two methods are available:

- **Obtain the Project ID by Calling an API**
- **Obtain the Project ID from the Console**

Obtain the Project ID by Calling an API

You can obtain the project ID by calling the IAM API used to query project information based on the specified criteria.

The API used to obtain a project ID is GET https://{Endpoint}/v3/projects. {Endpoint} is the IAM endpoint and can be obtained from **Regions and Endpoints**. For details about API authentication, see **Authentication**.

The following is an example response. The value of **id** is the project ID.

```
{
    "projects": [
        {
            "domain_id": "65ewtrgaggshhk1223245sghjlse684b",
            "is_domain": false,
            "parent_id": "65ewtrgaggshhk1223245sghjlse684b",
            "name": "project_name",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4adasfjljaaaakla12334jklga9sasfg"
            },
            "id": "a4adasfjljaaaakla12334jklga9sasfg",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

Obtain a Project ID from the Console

To obtain a project ID from the console, perform the following operations:

1. Log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list.

   On the **My Credentials** page, view the project ID (value in the **Project ID** column).

B Change History

Released On	Description
2024-04-15	• This issue is the first official release.

