Web 3.0 Node Engine Service (NES)

API Reference

Issue 01

Date 2023-08-07

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property 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 Cloud 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 Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Before You Start 1
1.1 Overview 1
1.2 Calling Method 1
1.3 Endpoints 1
1.4 Basic Concepts 1
2 API Overview 3
3 API Calling 5
3.1 API Requests 5
3.2 Authentication and Authorization 8
3.3 Responses 9
4 Examples 11
4.1 Example 1: Creating an Ethereum Mainnet Node 11
5 API 12
5.1 Node Management 12
5.1.1 Obtaining the Overview Information 12
5.1.2 Obtaining the Network Types 16
5.1.3 Obtaining All Specifications 20
5.1.4 Obtaining All Nodes on a Specified Network 25
5.1.5 Creating Nodes 30
5.1.6 Querying a Node 34
5.1.7 Deleting a Node 40
5.1.8 Changing the Node Specifications 44
5.1.9 Obtaining Available Specifications for Node Scaling 48
5.2 Node Monitoring 52
5.2.1 Obtaining the Node Monitoring Information 52
5.2.2 Obtaining Node Alarms 58
5.2.3 Obtaining the API Calling Information of a Node in a Specified Period 64
5.2.4 Obtaining the Status of a Staking Node in a Specified Period 69
5.3 API Keys 74
5.3.1 Creating an API Key 74
5.3.2 Obtaining All API Keys of a User 78
5.3.3 Deleting an API Key 82

5.4 Certificate Management....85
5.4.1 Downloading the Certificates....85
6 Permissions Policies and Supported Actions....90
6.1 Introduction....90
7 Appendix....92
7.1 Status Codes....92
7.2 Error Codes....95

1 Before You Start

1.1 Overview

Thanks for using Node Engine Service (NES). NES helps you quickly deploy, manage, and maintain a blockchain network in the cloud, reducing the threshold for using blockchains. With NES, you can focus on the development and innovation of your own business and quickly migrate your business to blockchains.

In addition to the web console, APIs are also available for managing your NES O&M resources and creating/deleting nodes.

If you plan to use an NES feature through an API, ensure that you are familiar with blockchain concepts.

1.2 Calling Method

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

1.3 Endpoints

An endpoint is the request address for calling an API. Endpoints vary depending on services and regions. For the endpoints of all services, see **Regions and Endpoints**.

1.4 Basic Concepts

- Account

  An account is created upon successful registration. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity and should not be used directly to perform routine management. For security purposes, create 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).

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

- Region

  Regions are divided from the dimensions of geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified as universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides services of the same type only or for specific tenants.

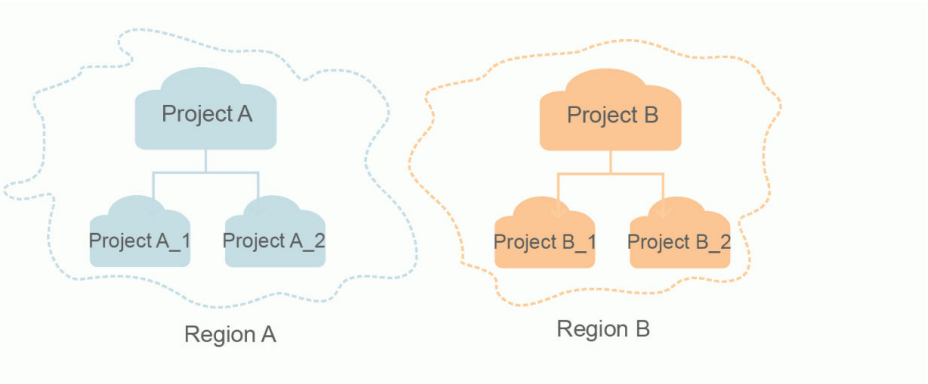
- Availability Zone (AZ)

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

- Project

  Default projects are defined to group and isolate resources (compute, storage, and network) across regions. You can grant users permissions to access resources in the region associated with a default project. For more refined access control, create subprojects under a project and create resources in the subprojects. Users can then be assigned permissions to access only specific resources in the subprojects.

  Figure 1-1 Project isolation model

  To view a project ID, go to the **My Credentials** page.

2 API Overview

You can use the APIs provided by NES to create nodes, create keys, and query a node. **Table 2-1** lists the APIs. For details about APIs, see **API**.

Table 2-1 API overview

Type	API	Description
NES	Deleting an API key	This API is used to delete an API key.
	Obtaining all API keys of a user	This API is used to obtain all API keys of a specified user.
	Creating an API key	This API is used to create an API key and return the API key file.
	Obtaining the network types	This API is used to obtain the network types.
	Obtaining all specifications	This API is used to obtain all specifications.
	Obtaining all nodes on a specified network	This API is used to obtain all nodes on a specified network.
	Creating nodes	This API is used to create nodes based on the specifications.
	Obtaining node alarms	This API is used to generate alarms according to the built-in alarm rules.
	Obtaining the API calling information of a node in a specified period	This API is used to obtain the API calling information of a node in a specified period.

Type	API	Description
	Obtaining available specifications for node scaling	This API is used to obtain available specifications for node scaling.
	Obtaining the node monitoring information	This API is used to obtain the node information, including the CPU usage, memory usage, uplink/downlink network traffic, storage, and disk read/write.
	Deleting a node	This API is used to delete a node.
	Querying a node	This API is used to query a node.
	Changing the node specifications	This API is used to change the node specifications.
	Obtaining the overview information	This API is used to obtain the overview information of the current user.
	Obtaining the status of a staking node in a specified period	This API is used to obtain the status of a staking node in a specified period, including its synchronization status, peer, and block height.
	Downloading the certificates	This API is used to download the certificates.

3 API Calling

3.1 API Requests

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 consists of the following parts:

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

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

- *URI-scheme*

  Protocol used to transmit requests. All APIs use HTTPS.

- *Endpoint*

  Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**.

  For example, the endpoint of NES in the AP-Singapore region is **bcs.ap-southeast-3.myhuaweicloud.com**.

- *resource-path*

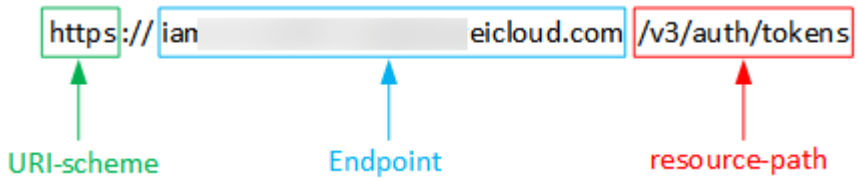
  API access path for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.

- *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.

For example, to obtain an IAM token in the AP-Singapore region, obtain the *Endpoint* of IAM (**iam.ap-southeast-3.myhuaweicloud.com**) for this region and

the *resource-path* (**/v3/auth/tokens**) in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:



NOTE

To simplify the URI display, 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 Method

HTTP-based request methods, which are also called operations or actions, specify the type of operations that you are requesting.

- **GET**: requests a server to return specified resources.
- **PUT**: requests a server to update specified resources.
- **POST**: requests a server to add resources or perform special operations.
- **DELETE**: requests a server to delete specified resources, for example, to delete an object.
- **HEAD**: requests a server resource header.
- **PATCH**: requests a server to update partial content of a specified resource. If the resource is unavailable, the PATCH method is used to create a resource.

For example, in the URI of the API for **obtaining a user token**, the request method is **POST**. The request is as follows:

```
POST https://iam.ap-southeast-3.myhuaweicloud.com/v3/auth/tokens
```

Request Header

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

Common request header fields are as follows:

- **Content-Type**: specifies the request body type or format. This field is mandatory and its default value is **application/json**. Other values of this field will be provided for specific APIs if any.
- **X-Auth-Token**: specifies a user token only for token-based API authentication. It is a response to the API for obtaining a user token. This API is the only one that does not require authentication.

> **NOTE**
>
> In addition to supporting token-based authentication, APIs also support authentication using access key ID/secret access key (AK/SK). During AK/SK-based authentication, an SDK is used to sign a request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request.
>
> For more information, see **AK/SK Authentication**.

The API for **obtaining a user token** does not require authentication, so that only the **Content-Type** field needs to be added:

```
POST https://iam.ap-southeast-3.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

Request Message Body

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

Request bodies vary with APIs. Some APIs do not require a 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* (account name), *********** (password), and *ap-southeast-3* (project name) with the actual values. For details, see **Regions and Endpoints**.

> **NOTE**
>
> - The **scope** parameter specifies where a token takes effect. 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 Through Password Authentication**.
> - For details about how to obtain the token for a non-Huawei Cloud account, see **Obtaining a User Token Through Password Authentication**.

```
POST https://iam.ap-southeast-3.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json

{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "ap-southeast-3"
            }
        }
    }
}
```


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 target user token. Then, this token can be used to authenticate the calling of other APIs.

3.2 Authentication and Authorization

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

- Token-based authentication: Requests are authenticated using a token.
- AK/SK authentication: Requests are encrypted using AK/SK pairs.

Token-based Authentication

NOTE

The validity period of a token is 24 hours. If a token is used for authentication, cache it to prevent frequent API calls.

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

When calling an API to obtain a user token, set **auth.scope** in the request body to **project**.

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

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

```
POST https://iam.ap-southeast-3.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```


AK/SK Authentication

NOTE

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

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

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

In AK/SK authentication, you can use an AK/SK to sign requests based on the signature algorithm or using the signing SDK. For details about how to sign requests and use the signing SDK, see **AK/SK Signing and Authentication Guide**.

NOTICE

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

3.3 Responses

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**.

If status code 201 is returned for the calling of the API for **obtaining a user token**, the request is successful.

Response Header

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

Figure 3-1 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. Then, this token can be used to authenticate the calling of other APIs.

Figure 3-1 Response header for the API used to obtain a user token

Response Body

The body of a response is often returned in 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**. The following describes part of the response body.

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

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

4 Examples

4.1 Example 1: Creating an Ethereum Mainnet Node

1. Call the IAM API to **obtain a user token**.
2. Call the API for **obtaining the network type** to choose a proper network type ID (*network_id*).
3. **Create a node**.

Request example

URI

https://*your_request_endpoint*/v1/*your_project_id*/node-provider/networks/*your_network_id*/nodes

Header

X-Auth-Token: *your_token*

Body

{

"cpu": 2,

"ram": 8192,

"node_type": "Full node",

"charge_mode": "postPaid",

"node_mode": "single",

"node_num": 1

}

5 API

5.1 Node Management

5.1.1 Obtaining the Overview Information

Description

This API is used to obtain the overview information of the current user.

URI

GET /v1/{project_id}/node-provider/summary

Table 5-1 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters

Request Parameters

Table 5-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Response Parameters

Status code: 200

Table 5-3 Response body parameters

Parameter	Type	Description
chain	Integer	Network type. Minimum value: 0 Maximum value: 2,147,483,647
node	Integer	Number of nodes. Minimum value: 0 Maximum value: 2,147,483,647
full_node	Integer	Number of full nodes. Minimum value: 0 Maximum value: 2,147,483,647
staking	Integer	Number of staking nodes. Minimum value: 0 Maximum value: 2,147,483,647
call	Integer	Number of calls. Minimum value: 0 Maximum value: 2,147,483,647
alarm	Integer	Number of alarms. Minimum value: 0 Maximum value: 2,147,483,647

Status code: 400

Table 5-4 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-5 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-6 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

None

Example Response

Status code: 200

Request successful.

```
{
  "chain" : 1,
  "node" : 2,
  "full_node" : 1,
  "staking" : 1,
  "call" : 10000,
  "alarm" : 10
}
```

Status code: 400

Request failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
```



```
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Request failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.1.2 Obtaining the Network Types

Description

This API is used to obtain the network types.

URI

GET /v1/{project_id}/node-provider/networks/type

Table 5-7 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters

Request Parameters

Table 5-8 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Response Parameters

Status code: 200

Table 5-9 Response body parameters

Parameter	Type	Description
[*Array elements*]	Array of **SupportedFramework** objects	List of supported network types.

Table 5-10 SupportedFramework

Parameter	Type	Description
name	String	Framework name. Minimum length: 0 characters Maximum length: 100 characters
networks	Array of **SupportedNetwork** objects	List of supported network types.

Table 5-11 SupportedNetwork

Parameter	Type	Description
id	String	Network type ID. Minimum length: 36 characters Maximum length: 36 characters

Parameter	Type	Description
name	String	Network name. Minimum length: 0 characters Maximum length: 100 characters

Status code: 400

Table 5-12 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-13 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters

Parameter	Type	Description
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-14 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

None

Example Response

Status code: 200

Request successful.

```
[ {
  "name" : "Ethereum",
  "networks" : [ {
    "id" : "468eda20-040b-11ee-877d-fa163e6c5c60",
    "name" : "Sepolia"
  } ]
} ]
```

Status code: 400

Request failed.

```
{
  "code" : 400,
```



```
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Request failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.1.3 Obtaining All Specifications

Description

This API is used to obtain all specifications.

URI

GET /v1/{project_id}/node-provider/networks/{network_id}/flavor

Table 5-15 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters
network_id	Yes	String	Network ID. Minimum length: 36 characters Maximum length: 36 characters

Table 5-16 Query parameters

Parameter	Mandatory	Type	Description
node_type	No	String	Node type, which can be **Full node** (default) or **Full node (Staking supported)**. Minimum length: 0 characters Maximum length: 36 characters

Request Parameters

Table 5-17 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Response Parameters

Status code: 200

Table 5-18 Response body parameters

Parameter	Type	Description
flavors	Array of **CreateNodeMachineFlavor** objects	All specifications.
count	Integer	Number of records. Minimum value: 0 Maximum value: 1000

Table 5-19 CreateNodeMachineFlavor

Parameter	Type	Description
cpu	Integer	Number of CPU cores. Minimum value: 1 Maximum value: 1024
ram	Integer	Memory. Minimum value: 1 Maximum value: 1,048,576
description	String	Description, which can be **Test**, **Preferred**, **Stable**, or **Powerful**. Minimum length: 1 character Maximum length: 50 characters
avail_node_mode	Array of strings	Availability mode of the node.

Status code: 400

Table 5-20 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters

Parameter	Type	Description
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-21 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-22 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters

Parameter	Type	Description
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

None

Example Response

Status code: 200

Request successful.

```
{
  "flavors" : [ {
    "cpu" : 2,
    "ram" : 8192,
    "description" : "Preferred",
    "avail_node_mode" : [ "single", "HA" ]
  } ],
  "count" : 1
}
```

Status code: 400

Request failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
```



```
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Request failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.1.4 Obtaining All Nodes on a Specified Network

Description

This API is used to obtain all nodes on a specified network.

URI

GET /v1/{project_id}/node-provider/networks/{network_id}/nodes

Table 5-23 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters
network_id	Yes	String	Network ID. Minimum length: 36 characters Maximum length: 36 characters

Table 5-24 Query parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Start position of pagination query. Minimum value: 0 Maximum value: 1000
limit	No	Integer	Number of items returned on each page. Minimum value: 1 Maximum value: 1000

Request Parameters

Table 5-25 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Response Parameters

Status code: 200

Table 5-26 Response body parameters

Parameter	Type	Description
nodes	Array of **Node** objects	Node list information.
count	Integer	Number of records. Minimum value: 0 Maximum value: 1000

Table 5-27 Node

Parameter	Type	Description
id	String	Node ID. Minimum length: 36 characters Maximum length: 45 characters
node_type	String	Node type, which can be **Full node** or **Full node (Staking supported)**. Minimum length: 0 characters Maximum length: 100 characters
flavor	String	Node specifications. Minimum length: 0 characters Maximum length: 100 characters
create_time	String	Time when the node was created. Minimum length: 0 characters Maximum length: 100 characters
status	Object	Node status, which can be **Available**, **Unavailable**, **Creating**, or **Upgrading**.

Status code: 400

Table 5-28 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-29 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-30 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

None

Example Response

Status code: 200

Request successful.

```
{
  "nodes" : [ {
    "id" : "468eda20-040b-11ee-877d-fa163e6c5c60",
    "node_type" : "Full node",
    "flavor" : {
      "cpu" : 4,
      "ram" : 16384,
      "description" : "Preferred"
    },
    "create_time" : "2023-06-06T09:41:37.000553+08:00",
    "status" : "Available"
  } ],
  "count" : 1
}
```

Status code: 400

Verify parameter failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.

Status Code	Description
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.1.5 Creating Nodes

Description

This API is used to create nodes based on the specifications.

URI

POST /v1/{project_id}/node-provider/networks/{network_id}/nodes

Table 5-31 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters
network_id	Yes	String	Network ID. Minimum length: 36 characters Maximum length: 36 characters

Request Parameters

Table 5-32 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Table 5-33 Request body parameters

Parameter	Mandatory	Type	Description
cpu	Yes	Long	Number of CPU cores. Minimum value: 1 Maximum value: 1024
ram	Yes	Integer	RAM size. Minimum value: 1 Maximum value: 1,048,576
node_type	Yes	String	Node type, which can be **Full node** or **Full node (Staking supported)**. Minimum length: 0 characters Maximum length: 100 characters
charge_mode	Yes	String	Billing mode, which can be **postPaid** or **prePaid**. Minimum length: 0 characters Maximum length: 100 characters
node_mode	Yes	String	Node mode, which can be **single** or **HA**. Minimum length: 0 characters Maximum length: 100 characters
node_num	Yes	Integer	Number of nodes. A maximum of five nodes can be created at a time. Minimum value: 1 Maximum value: 5
mev_boost	No	String	MEV-Boost address. If this parameter is left empty, MEV-Boost connection is disabled. Minimum length: 0 characters Maximum length: 100 characters

Response Parameters

Status code: 200

Table 5-34 Response body parameters

Parameter	Type	Description
jobs	Array of strings	Task list.

Status code: 400

Table 5-35 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-36 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters

Parameter	Type	Description
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-37 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

```
{
  "cpu" : 2,
  "ram" : 8192,
  "node_type" : "Full node",
  "charge_mode" : "postPaid",
  "node_mode" : "single",
  "node_num" : 1
}
```

Example Response

Status code: 200

Request successful.

```
{
  "jobs" : [ "c85f549f-7c80-11ed-b1f5-0242a9fe1e03" ]
}
```

Status code: 400

Verify parameter failed.


```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.1.6 Querying a Node

Description

This API is used to query a node.

URI

GET /v1/{project_id}/node-provider/nodes/{node_id}

Table 5-38 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters
node_id	Yes	String	Node ID. Minimum length: 36 characters Maximum length: 45 characters

Request Parameters

Table 5-39 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Response Parameters

Status code: 200

Table 5-40 Response body parameters

Parameter	Type	Description
id	String	Node ID. Minimum length: 36 characters Maximum length: 45 characters
ws_address	String	WebSocket terminal address. Minimum length: 0 characters Maximum length: 100 characters
http_address	String	HTTP terminal address. Minimum length: 0 characters Maximum length: 100 characters

Parameter	Type	Description
grpc_address	String	gRPC terminal address. Minimum length: 0 characters Maximum length: 100 characters
flavor	**MachineFlavor** object	Node specifications.
status	String	Node status, which can be **Available**, **Unavailable**, or **Upgrading**. Minimum length: 0 characters Maximum length: 100 characters
network_type	**NetworkType** object	Node network information.
node_type	String	Node type, which can be **Full node** or **Full node (Staking supported)**. Minimum length: 0 characters Maximum length: 100 characters
high_availability	Boolean	Whether HA is available.
create_time	String	Time when the node was created. Minimum length: 0 characters Maximum length: 100 characters
mev_boost_address	String	MEV-Boost Address Minimum length: 0 characters Maximum length: 100 characters
mev_boost_status	Boolean	Whether MEV-Boost is normal.

Table 5-41 MachineFlavor

Parameter	Type	Description
cpu	Integer	Number of CPU cores. Minimum value: 1 Maximum value: 1024
ram	Integer	Memory. Minimum value: 1 Maximum value: 1,048,576

Parameter	Type	Description
description	String	Description, which can be **Test**, **Preferred**, **Stable**, or **Powerful**. Minimum length: 1 character Maximum length: 50 characters

Table 5-42 NetworkType

Parameter	Type	Description
id	String	Network type ID. Minimum length: 0 characters Maximum length: 1000 characters
framework	String	Public blockchain name. Minimum length: 0 characters Maximum length: 1000 characters
name	String	Network name. Minimum length: 0 characters Maximum length: 1000 characters

Status code: 400

Table 5-43 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-44 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-45 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

None

Example Response

Status code: 200

Request successful.


```
{
  "id" : "468eda20-040b-11ee-877d-fa163e6c5c60",
  "ws_address" : "ws://100.100.100.100",
  "http_address" : "http://100.100.100.100",
  "grpc_address" : "100.100.100.100:30002",
  "flavor" : {
    "cpu" : 2,
    "ram" : 8196,
    "description" : "Preferred"
  },
  "status" : "Available",
  "network_type" : {
    "id" : "79d0f3b9-8ce1-11ed-8398-0242a9fe1e02",
    "framework" : "Ethereum",
    "name" : "Goerli"
  },
  "node_type" : "Full node",
  "high_availability" : false,
  "create_time" : "2023-06-06T09:41:37.000553+08:00",
  "mev_boost_address" : "123.123.123.123:1234",
  "mev_boost_status" : true
}
```


Status code: 400

Verify parameter failed.


```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```


Status code: 401

Authentication failed.


```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```


Status code: 500

Internal service error.


```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```


Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.1.7 Deleting a Node

Description

This API is used to delete a node.

URI

DELETE /v1/{project_id}/node-provider/nodes/{node_id}

Table 5-46 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters
node_id	Yes	String	Node ID. Minimum length: 36 characters Maximum length: 45 characters

Request Parameters

Table 5-47 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Response Parameters

Status code: 200

Table 5-48 Response body parameters

Parameter	Type	Description
jobs	Array of strings	Task list.

Status code: 400

Table 5-49 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-50 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-51 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

None

Example Response

Status code: 200

Request successful.

```
{
  "jobs" : [ "c85f549f-7c80-11ed-b1f5-0242a9fe1e03" ]
}
```

Status code: 400

Verify parameter failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.1.8 Changing the Node Specifications

Description

This API is used to change the node specifications.

URI

PUT /v1/{project_id}/node-provider/nodes/{node_id}

Table 5-52 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters
node_id	Yes	String	Node ID. Minimum length: 36 characters Maximum length: 45 characters

Request Parameters

Table 5-53 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Table 5-54 Request body parameters

Parameter	Mandatory	Type	Description
cpu	Yes	Integer	New CPU specifications. Minimum value: 1 Maximum value: 1024

Parameter	Mandatory	Type	Description
ram	Yes	Integer	New RAM specifications. Minimum value: 1 Maximum value: **100**

Response Parameters

Status code: 200

Table 5-55 Response body parameters

Parameter	Type	Description
jobs	Array of strings	Task list.

Status code: 400

Table 5-56 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-57 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-58 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

```
{
  "cpu" : 2,
  "ram" : 8196
}
```


Example Response

Status code: 200

Request successful.

```
{
  "jobs" : [ "c85f549f-7c80-11ed-b1f5-0242a9fe1e03" ]
}
```

Status code: 400

Verify parameter failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.1.9 Obtaining Available Specifications for Node Scaling

Description

This API is used to obtain available specifications for node scaling.

URI

GET /v1/{project_id}/node-provider/nodes/{node_id}/flavor

Table 5-59 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters
node_id	Yes	String	Node ID. Minimum length: 36 characters Maximum length: 45 characters

Request Parameters

Table 5-60 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Response Parameters

Status code: 200

Table 5-61 Response body parameters

Parameter	Type	Description
flavors	Array of **MachineFlavor** objects	All specifications.
count	Integer	Number of records. Minimum value: 0 Maximum value: 1000

Table 5-62 MachineFlavor

Parameter	Type	Description
cpu	Integer	Number of CPU cores. Minimum value: 1 Maximum value: 1024
ram	Integer	Memory. Minimum value: 1 Maximum value: 1,048,576
description	String	Description, which can be **Test**, **Preferred**, **Stable**, or **Powerful**. Minimum length: 1 character Maximum length: 50 characters

Status code: 400

Table 5-63 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters

Parameter	Type	Description
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-64 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-65 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters

Parameter	Type	Description
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

None

Example Response

Status code: 200

Request successful.

```
{
  "flavors" : [ {
    "cpu" : 4,
    "ram" : 16384,
    "description" : "Preferred"
  } ],
  "count" : 1
}
```

Status code: 400

Request failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```


Status Code

Status Code	Description
200	Request successful.
400	Request failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.2 Node Monitoring

5.2.1 Obtaining the Node Monitoring Information

Description

This API is used to obtain the node information, including the CPU usage, memory usage, uplink/downlink network traffic, storage, and disk read/write.

URI

POST /v1/{project_id}/node-provider/nodes/{node_id}/metrics

Table 5-66 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters
node_id	Yes	String	Node ID. Minimum length: 36 characters Maximum length: 45 characters

Request Parameters

Table 5-67 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Table 5-68 Request body parameters

Parameter	Mandatory	Type	Description
metric_names	No	Array of strings	Metrics. Options: **cpuUsage**: CPU usage. **diskUsedRate**: disk usage. **memUsedRate**: physical memory usage. **sendBytesRate**: uplink rate (byte/s). **recvBytesRate**: downlink rate (byte/s). **cpuCoreLimit**: total CPU cores. **cpuCoreUsed**: used CPU cores. **totalMem**: total physical memory. **freeMem**: available physical memory. **diskCapacity**: disk capacity. **diskAvailableCapacity**: available disk space. By default, the first five metrics are used.
period	Yes	Integer	Monitoring data granularity. For example, **60** indicates 1 minute and **300** indicates 5 minutes. Minimum value: 0 Maximum value: 86,400
timerange	Yes	String	Query time range. For example, **-1.-1.60** indicates the latest 60 minutes (startTimeInMillis.endTimeInMillis.durationInMinutes). Minimum length: 0 characters Maximum length: 100 characters

Parameter	Mandatory	Type	Description
statistics	Yes	Array of strings	Statistical mode, which can be **maximum**, **minimum**, **sum**, **average**, or **sampleCount**.

Response Parameters

Status code: 200

Table 5-69 Response body parameters

Parameter	Type	Description
metrics	Array of **MetricItemResultAPI** objects	Metric list.

Table 5-70 MetricItemResultAPI

Parameter	Type	Description
metric	**MetricDemision** object	Metric description.
dataPoints	Array of **MetricDataPoints** objects	Basic information of the monitoring data.

Table 5-71 MetricDemision

Parameter	Type	Description
namespace	String	Namespace. Minimum length: 1 character Maximum length: 65,535 characters
metricName	String	Metric name. Minimum length: 1 character Maximum length: 65,535 characters
dimensions	Array of **Dimension** objects	Dimension list.

Table 5-72 Dimension

Parameter	Type	Description
name	String	Dimension name. Minimum length: 1 character Maximum length: 65,535 characters
value	String	Dimension value. Minimum length: 1 character Maximum length: 65,535 characters

Table 5-73 MetricDataPoints

Parameter	Type	Description
timestamp	String	Timestamp. Minimum length: 0 characters Maximum length: 100 characters
unit	String	Metric unit. Minimum length: 1 character Maximum length: 65,535 characters
statistics	Array of **StatisticValue** objects	Statistical mode.

Table 5-74 StatisticValue

Parameter	Type	Description
statistic	String	Statistical mode. Minimum length: 1 character Maximum length: 65,535 characters
value	Double	Statistics result. Minimum value: 0 Maximum value: 2,147,483,647

Status code: 400

Table 5-75 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-76 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-77 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

```
{
  "metric_names" : [ "cpuUsage" ],
  "period" : 300,
  "timerange" : "-1.-1.60",
  "statistics" : [ "maximum" ]
}
```

Example Response

Status code: 200

Request successful.

```
{
  "metrics" : [ {
    "metric" : {
      "namespace" : "123",
      "metricName" : "cpuUsage",
      "dimensions" : [ {
        "name" : "nodeIP",
        "value" : "192.168.0.1"
      } ]
    },
    "dataPoints" : [ {
      "timestamp" : 1686059700000,
      "unit" : "Percent",
      "statistics" : [ {
        "statistic" : "average",
        "value" : 20
      } ]
    } ]
  } ]
}
```

Status code: 400

Verify parameter failed.


```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.2.2 Obtaining Node Alarms

Description

This API is used to generate alarms according to the built-in alarm rules.

URI

POST /v1/{project_id}/node-provider/nodes/{node_id}/alarms/{alert_type}

Table 5-78 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters
node_id	Yes	String	Node ID. Minimum length: 36 characters Maximum length: 45 characters
alert_type	Yes	String	Query type. **active_alert** indicates that active alarms are queried and **history_alert** indicates that historical alarms are queried. Minimum length: 1 character Maximum length: 15 characters

Request Parameters

Table 5-79 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Table 5-80 Request body parameters

Parameter	Mandatory	Type	Description
time_range	Yes	String	Time range specified to query data of the last N minutes when the client time is inconsistent with the server time. It can also be used to accurately query data in a specified period. For example, **-1.-1.30** indicates the latest 30 minutes. Minimum length: 0 characters Maximum length: 100 characters
sort	No	**Sort** object	Sorting mode of the list.

Table 5-81 Sort

Parameter	Mandatory	Type	Description
order_by	No	Array of strings	List of sorted fields.
order	No	String	Sorting order, which can be **asc** (ascending order) or **desc** (descending order). Minimum length: 0 characters Maximum length: 100 characters

Response Parameters

Status code: 200

Table 5-82 Response body parameters

Parameter	Type	Description
data	Array of **AlarmInfo** objects	Alarm list.
count	Integer	Number of records. Minimum value: 0 Maximum value: 1000

Table 5-83 AlarmInfo

Parameter	Type	Description
id	String	Event or alarm ID, which is automatically generated by the system. Minimum length: 0 characters Maximum length: 100 characters
starts_at	Integer	Time when an event or alarm is generated. The value is a CST timestamp precise down to the millisecond. Minimum value: 0 Maximum value: 18,446,744,073,709,551,616
ends_at	Integer	Time when an event or alarm is cleared. The value is a CST timestamp precise down to the millisecond. If the value is **0**, the event or alarm is not deleted. Minimum value: 0 Maximum value: 18,446,744,073,709,551,616
timeout	Integer	Duration at which an alarm is automatically cleared, in millisecond. For example, if the duration is one minute, set this parameter to **60,000**. The default value is 3 days, that is, 3 days x 24 hours x 60 minutes x 1000 ms = **4,320,000** ms. Minimum value: 0 Maximum value: 18,446,744,073,709,551,616
metadata	Object	Detailed information about an event or alarm.
annotations	Object	Alarm details.

Status code: 400

Table 5-84 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters

Parameter	Type	Description
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-85 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-86 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters

Parameter	Type	Description
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

```
{
  "time_range" : "-1.-1.30",
  "sort" : {
    "order_by" : [ "starts_at" ],
    "order" : "desc"
  }
}
```

Example Response

Status code: 200

Request successful.

```
{
  "data" : [ {
    "id" : "3512004942693961460",
    "starts_at" : 1671518362723,
    "ends_at" : 1671518362723,
    "timeout" : 1440000,
    "metadata" : "string",
    "annotations" : "string"
  } ],
  "count" : 1
}
```

Status code: 400

Verify parameter failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.2.3 Obtaining the API Calling Information of a Node in a Specified Period

Description

This API is used to obtain the API calling information of a node in a specified period.

URI

POST /v1/{project_id}/node-provider/nodes/{node_id}/api-statistics

Table 5-87 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters

Parameter	Mandatory	Type	Description
node_id	Yes	String	Node ID. Minimum length: 36 characters Maximum length: 45 characters

Request Parameters

Table 5-88 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Table 5-89 Request body parameters

Parameter	Mandatory	Type	Description
time_range	Yes	String	Query time range. For example, **-1.-1.5** indicates the latest 5 minutes. Minimum length: 0 characters Maximum length: 100 characters
period	Yes	Integer	Monitoring data granularity. For example, **60** indicates 1 minute and **300** indicates 5 minutes. Minimum value: 0 Maximum value: 86,400
statistics	No	Array of strings	Statistical mode. Minimum length: 0 characters Maximum length: 100 characters

Response Parameters

Status code: 200

Table 5-90 Response body parameters

Parameter	Type	Description
top	Array of **TopMethodResult** objects	List of top 5 requests.
data	Array of **MethodResult** objects	API calling data.

Table 5-91 TopMethodResult

Parameter	Type	Description
method	String	Method name. Minimum length: 0 characters Maximum length: 100 characters
count	Integer	Number of records. Minimum value: 0 Maximum value: 2,147,483,648

Table 5-92 MethodResult

Parameter	Type	Description
timestamp	String	Timestamp. Minimum length: 0 characters Maximum length: 100 characters
count	Integer	Number of calls. Minimum value: 0 Maximum value: 2,147,483,648

Status code: 400

Table 5-93 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters

Parameter	Type	Description
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-94 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-95 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters

Parameter	Type	Description
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

```
{
  "time_range" : "-1.-1.5",
  "period" : 300,
  "statistics" : [ "average" ]
}
```

Example Response

Status code: 200

Request successful.

```
{
  "top" : [ {
    "method" : "geth",
    "count" : 10
  } ],
  "data" : [ {
    "timestamp" : "2023-06-06T22:00:00+08:00",
    "count" : 10
  } ]
}
```

Status code: 400

Verify parameter failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```


Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.2.4 Obtaining the Status of a Staking Node in a Specified Period

Description

This API is used to obtain the status of a staking node in a specified period, including its synchronization status, peer, and block height.

URI

POST /v1/{project_id}/node-provider/nodes/{node_id}/status

Table 5-96 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters

Parameter	Mandatory	Type	Description
node_id	Yes	String	Node ID. Minimum length: 36 characters Maximum length: 45 characters

Request Parameters

Table 5-97 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Table 5-98 Request body parameters

Parameter	Mandatory	Type	Description
time_range	Yes	String	Query time range. For example, **-1.-1.30** indicates the latest 30 minutes. Minimum length: 0 characters Maximum length: 100 characters

Response Parameters

Status code: 200

Table 5-99 Response body parameters

Parameter	Type	Description
node_status	Array of **BeaconNode Status** objects	Node status.
count	Integer	Number of nodes. Minimum value: 0 Maximum value: 1000

Table 5-100 BeaconNodeStatus

Parameter	Type	Description
node_info	Array of **NodeInfoResult** objects	Number of inbound and outbound peers.

Table 5-101 NodeInfoResult

Parameter	Type	Description
timestamp	String	Timestamp. Minimum length: 0 characters Maximum length: 100 characters
inbound	Integer	Number of inbound peers. Minimum value: 0 Maximum value: 10,000
outbound	Integer	Number of outbound peers. Minimum value: 0 Maximum value: 10,000
block_height	Integer	Block height. Minimum value: 0 Maximum value: 2,147,483,647

Status code: 400

Table 5-102 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters

Parameter	Type	Description
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-103 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-104 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters

Parameter	Type	Description
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

```
{
  "time_range" : "-1.-1.30"
}
```

Example Response

Status code: 200

Request successful.

```
{
  "node_status" : [ {
    "node_info" : [ {
      "timestamp" : "2023-05-09T10:00:24.775Z",
      "inbound" : 45,
      "outbound" : 32,
      "block_height" : 17222778
    } ]
  } ],
  "count" : 1
}
```

Status code: 400

Verify parameter failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
```



```
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.3 API Keys

5.3.1 Creating an API Key

Description

This API is used to create an API key and return the API key file.

URI

POST /v1/{project_id}/node-provider/credentials

Table 5-105 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters

Request Parameters

Table 5-106 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Table 5-107 Request body parameters

Parameter	Mandatory	Type	Description
description	No	String	API key description. Minimum length: 0 characters Maximum length: 100 characters

Response Parameters

Status code: 400

Table 5-108 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-109 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-110 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

```
{
"description": "For test"
}
```


Example Response

Status code: 200

Request successful.

```
PKXXX
```

Status code: 400

Verify parameter failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.3.2 Obtaining All API Keys of a User

Description

This API is used to obtain all API keys of a specified user.

URI

GET /v1/{project_id}/node-provider/credentials

Table 5-111 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters

Table 5-112 Query parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Start position of pagination query. Minimum value: 0 Maximum value: 1000
limit	No	Integer	Number of items returned on each page. Minimum value: 1 Maximum value: 1000

Request Parameters

Table 5-113 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Response Parameters

Status code: 200

Table 5-114 Response body parameters

Parameter	Type	Description
credentials	Array of **Credential** objects	List of API keys.
count	Integer	Number of records. Minimum value: 0 Maximum value: 1000

Table 5-115 Credential

Parameter	Type	Description
id	String	API key ID. Minimum length: 36 characters Maximum length: 36 characters
description	String	API key description. Minimum length: 1 character Maximum length: 100 characters
create_time	String	Time when the API key was created. Minimum length: 0 characters Maximum length: 100 characters
update_time	String	Time when the credential was used for the last time. Minimum length: 0 characters Maximum length: 100 characters

Status code: 400

Table 5-116 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters

Parameter	Type	Description
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-117 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-118 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters

Parameter	Type	Description
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

None

Example Response

Status code: 200

Request successful.

```
{
  "credentials" : [ {
    "id" : "3bca624c-0468-11ee-8322-0255ac100043",
    "description" : "test",
    "create_time" : "2023-06-06T20:47:02.15507+08:00",
    "update_time" : "2023-06-06T20:47:02.15507+08:00"
  } ],
  "count" : 1
}
```

Status code: 400

Verify parameter failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.


```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.3.3 Deleting an API Key

Description

This API is used to delete an API key.

URI

DELETE /v1/{project_id}/node-provider/credentials/{credential_id}

Table 5-119 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters
credential_id	Yes	String	API key ID. Minimum length: 36 characters Maximum length: 36 characters

Request Parameters

Table 5-120 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Response Parameters

Status code: 400

Table 5-121 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-122 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters

Parameter	Type	Description
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-123 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

None

Example Response

Status code: 200

Request successful.

```
null
```


Status code: 400

Verify parameter failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.

```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

5.4 Certificate Management

5.4.1 Downloading the Certificates

Description

This API is used to download the certificates.

URI

GET /v1/{project_id}/node-provider/nodes/{node_id}/certs

Table 5-124 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum length: 32 characters Maximum length: 32 characters
node_id	Yes	String	Node ID. Minimum length: 36 characters Maximum length: 45 characters

Table 5-125 Query parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Start position of pagination query. Minimum value: 0 Maximum value: 1000
limit	No	Integer	Number of items returned on each page. Minimum value: 1 Maximum value: 1000

Request Parameters

Table 5-126 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Minimum length: 1 character Maximum length: 65,535 characters

Response Parameters

Status code: 400

Table 5-127 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 401

Table 5-128 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Status code: 500

Table 5-129 Response body parameters

Parameter	Type	Description
code	String	Status code. Minimum length: 3 characters Maximum length: 3 characters
error_code	String	Error code. Minimum length: 12 characters Maximum length: 12 characters
error_msg	String	Error message. Minimum length: 1 character Maximum length: 65,535 characters
message	String	Error details. Minimum length: 0 characters Maximum length: 65,535 characters

Example Request

None

Example Response

Status code: 200

Request successful.

```
PKXXX
```

Status code: 400

Verify parameter failed.

```
{
  "code" : 400,
  "error_code" : "BCS.03400001",
  "error_msg" : "Invalid request.",
  "message" : "unmarshal request data error"
}
```

Status code: 401

Authentication failed.

```
{
  "code" : 401,
  "error_code" : "BCS.03401001",
  "error_msg" : "Authorization failed.",
  "message" : "request token is not valid"
}
```

Status code: 500

Internal service error.


```
{
  "code" : 500,
  "error_code" : "BCS.03500001",
  "error_msg" : "Internal error.",
  "message" : "project [xxx] node [xxx] get service bearer user token error"
}
```

Status Code

Status Code	Description
200	Request successful.
400	Verify parameter failed.
401	Authentication failed.
500	Internal service error.

Error Code

For details, see **Error Codes**.

6 Permissions Policies and Supported Actions

6.1 Introduction

You can use IAM to implement fine-grained permissions management for your NES resources. If your Huawei Cloud account does not need individual IAM users, then you may skip this section.

A policy is a set of permissions defined in JSON format. By default, new IAM users do not have any permissions assigned. You need to add a user to one or more groups, and assign permissions policies to these groups. The user then inherits permissions from the groups it is a member of. This process is called authorization. After authorization, the user can perform specified operations on NES based on the permissions.

You can grant users permissions by using **roles** and **policies**. Roles are provided by IAM to define service-based permissions that match users' job responsibilities. Policie are more fine-grained, API-based permissions required to perform operations on specific cloud resources under certain conditions, meeting requirements for secure access control.

For details about the NES policies, see **Permissions Management**.

NOTE

Policy-based authorization is useful if you want to allow or deny the access to an API.

Supported Actions

Actions supported by policies are specific to APIs. Common concepts related to policies include:

- Permissions: statements in a policy that allow or deny certain operations
- APIs: APIs that will be called for performing certain operations.
- Actions: specific operations that are allowed or denied
- Dependencies: actions which a specific action depends on. When allowing an action for a user, you also need to allow its dependent actions for that user.

- Supported: IAM projects and enterprise projects

  Type of projects in which policies can be used to grant permissions. A policy can be applied to IAM projects, 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 only contain actions supporting IAM projects can be assigned to user groups and only take effect for IAM. Such policies will not take effect if they are assigned to user groups in Enterprise Management. For details about the differences between IAM and enterprise projects, see **What Are the Differences Between IAM and Enterprise Management?**

7 Appendix

7.1 Status Codes

Table 7-1 lists the status codes.

Table 7-1 Status codes

Status Code	Message	Description
100	Continue	The client should continue with its request. This code is issued on a provisional basis while request processing continues. It alerts the client to wait for a final response.
101	Switching Protocols	The protocol should be switched to a newer protocol. For example, the current HTTPS protocol should be switched to a later version.
200	Success	The request is successfully sent.
201	Created	The request for creating resources has been fulfilled.
202	Accepted	The request has been accepted, but the processing has not been completed.
203	Non-Authoritative Information	The server successfully processed the request, but is returning information that may be from another source.
204	NoContent	The request has been fulfilled, but the HTTPS response does not contain a response body. The status code is returned in response to an HTTPS **OPTIONS** request.

Status Code	Message	Description
205	Reset Content	The server has fulfilled the request, but the requester is required to reset the content.
206	Partial Content	The server has successfully processed a part of a **GET** request.
300	Multiple Choices	There are multiple options for the requested resource. The response contains a list of resource characteristics and addresses from which a user terminal (such as a browser) can choose the most appropriate one.
301	Moved Permanently	The requested resource has been assigned with a new permanent URI. This new URI is contained in the response.
302	Found	The requested resource resides temporarily under a different URI.
303	See Other	The response to the request can be found under a different URI. It should be retrieved using a **GET** or **POST** method.
304	Not Modified	The requested resource has not been modified. When the server returns this status code, it does not return any resources.
305	Use Proxy	The requested resource must be accessed through a proxy.
306	Unused	The HTTPS status code is no longer used.
400	BadRequest	The request is invalid. The client should not repeat the request without modifications.
401	Unauthorized	The authorization information provided by the client is incorrect or invalid.
402	Payment Required	This status code is reserved for future use.
403	Forbidden	The request is rejected. The server understands the request, but refuses to fulfill it. The client should not repeat the request without modifications.
404	NotFound	The requested resource could not be found. The client should not repeat the request without modifications.

Status Code	Message	Description
405	MethodNotAllowed	A request method is not supported for the requested resource. The client should not repeat the request without modifications.
406	Not Acceptable	The server could not fulfil the request according to the content characteristics of the request.
407	Proxy Authentication Required	This status code is similar to 401, but the client must first authenticate itself with the proxy.
408	Request Time-out	The client does not produce a request within the time that the server was prepared to wait. The client may repeat the request without modifications later.
409	Conflict	The request cannot be processed due to a conflict. The resource that the client attempts to create already exists, or the update request fails to be processed because of a conflict.
410	Gone	The requested resource is no longer available. The status code indicates that the requested resource has been deleted permanently.
411	Length Required	The server refuses to process the request without a defined **Content-Length**.
412	Precondition Failed	The server does not meet one of the preconditions that the requester puts on the request.
413	Request Entity Too Large	The server refuses to process a request because the request entity is too large. The server may disable the connection to prevent the client from sending requests consecutively. If the server cannot process the request temporarily, the response will contain a **Retry-After** field.
414	Request-URI Too Large	The request URI is too long for the server to process.
415	Unsupported Media Type	The server cannot process the media format in the request.
416	Requested range not satisfiable	The requested range is invalid.
417	Expectation Failed	The server fails to meet the requirements of the **Expect** request header field.

Status Code	Message	Description
422	UnprocessableEntity	The request was well-formed but was unable to be followed due to semantic errors.
429	TooManyRequests	The client has sent more requests than its rate limit is allowed within a given time, or the server has received more requests than it is able to process within a given time. In this case, the client should repeat requests after the time specified in the **Retry-After** header of the response expires.
500	InternalServerError	The server is able to receive the request but unable to understand the request.
501	Not Implemented	The server does not support the requested function.
502	Bad Gateway	The server was acting as a gateway or proxy and received an invalid request from a remote server.
503	ServiceUnavailable	The requested service is invalid. The client should not repeat the request without modifications.
504	ServerTimeout	The request cannot be fulfilled within a given time. This status code is returned to the client only when the **Timeout** parameter is specified in the request.
505	HTTPS Version not supported	The server does not support the HTTPS protocol version used in the request.

7.2 Error Codes

Table 7-2 lists the error codes.

Table 7-2 Error codes

Error Code	Error Message	Description	Solution
BCS.03400001	Invalid request.	The request is invalid.	Check the request information.
BCS.03400002	Insufficient node quota	The node quota is insufficient.	Increase the resource quota of the account.

Error Code	Error Message	Description	Solution
BCS.03400006	Content type not supported.	The current content type is not supported.	Check the request information.
BCS.03400020	Version not supported.	The current version is not supported.	Check the version information.
BCS.03400022	No available flavors for nodes.	No node configuration is available.	Select another available configuration.
BCS.03400023	Current nodeType does not support.	The selected node type is not supported.	Select another available type.
BCS.03401001	Authorization failed.	Authentication failed.	Check the authentication information.
BCS.03403002	No OBT qualification.	No permissions granted for this closed beta test.	Contact related personnel.
BCS.03404001	Resource not found.	Resources not found.	Check whether the resource information is correct.
BCS.03429001	Reach maximum parallel tasks.	The maximum number of concurrent tasks is reached.	Wait until the task is complete and try again.
BCS.03500001	Internal error.	Internal error.	Contact technical support engineers.

