Dedicated Distributed Storage Service

API Reference

Issue	01
Date	2023-11-17

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Security Declaration

Vulnerability

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

Contents

1 Before You Start 1
1.1 Overview 1
1.2 API Calling 1
1.3 Endpoints 1
1.4 Concepts 2
2 API Overview 4
3 Calling APIs 5
3.1 Making an API Request 5
3.2 Authentication 9
3.3 Response 11
4 API Description 13
4.1 Obtaining Details of a DSS Storage Pool 13
4.2 Obtaining Details of DSS Storage Pools 15
4.3 Querying Details About All Disks 19
4.4 Obtaining Details of API Versions 19
4.5 Obtaining Details of an API Version 21
5 Permissions Policies and Supported Actions 24
5.1 Introduction 24
5.2 DSS Storage Pool Management 25
5.3 Disk Management 25
A Appendixes 26
A.1 DSS Storage Pool Status 26
A.2 Error Codes 26
A.3 Status Codes 28
A.4 Obtaining a Project ID 29
B Change History 31

1 Before You Start

1.1 Overview

Welcome to *Dedicated Distributed Storage Service API Reference*. Dedicated Distributed Storage Service (DSS) provides you with dedicated storage pools which are physically isolated from other pools to ensure high security. With data redundancy and cache acceleration technologies, DSS delivers highly reliable, durable, low-latency, and stable storage resources. By flexibly interconnecting with various compute services, such as Dedicated Computing Cluster (DCC), Elastic Cloud Server (ECS), and Bare Metal Server (BMS), DSS is perfect for different scenarios, including high performance computing (HPC), online analytical processing (OLAP), and mixed loads.

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

If you plan to access DSS through an API, ensure that you are familiar with DSS concepts. For details, see **Service Overview**.

After the storage pool is deployed and becomes available, you need to create disks in the storage pool. For details about disk APIs, see **Elastic Volume Service API Reference**.

1.2 API Calling

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

1.3 Endpoints

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

1.4 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, which should not be used directly to perform routine management. For security purposes, create Identity and Access Management (IAM) users and grant them permissions for routine management.

- User

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

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

- Region

  Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

  For details, see **Region and AZ**.

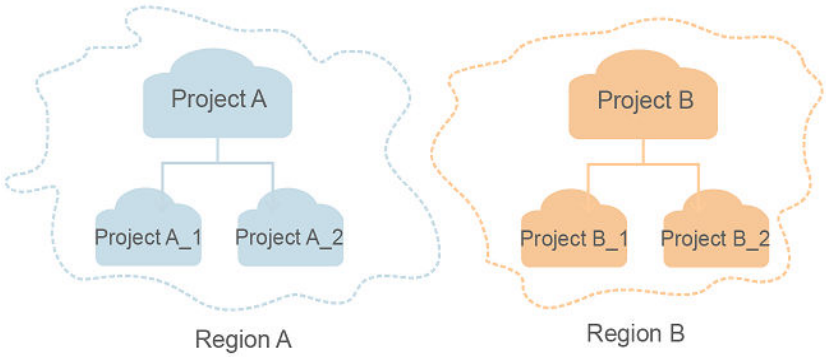
- AZ

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

- Project

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

Figure 1-1 Project isolation model



- Enterprise project

  Enterprise projects group and manage resources across regions. Resources in different enterprise projects are logically isolated. An enterprise project can contain resources of multiple regions, and resources can be added to or removed from enterprise projects.

  For details about enterprise projects and about how to obtain enterprise project IDs, see ***Enterprise Management User Guide***.

2 API Overview

Table 2-1 provides an overview of the DSS APIs.

Table 2-1 API overview

API	Description
Obtaining Details of a DSS Storage Pool	Obtain the details of a specified DSS storage pool, including the pool name, ID, capacity, type, and creation time.
Obtaining Details of DSS Storage Pools	Obtain the DSS storage pools requested by the tenant, including the pool names, IDs, capacities, types, and creation time. Filter query and pagination query are supported.

For details about disk APIs, see **Elastic Volume Service API Reference**.

3 Calling APIs

3.1 Making an API Request

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

Request URI

A request URI is in the following format:

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

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

Table 3-1 URI parameter description

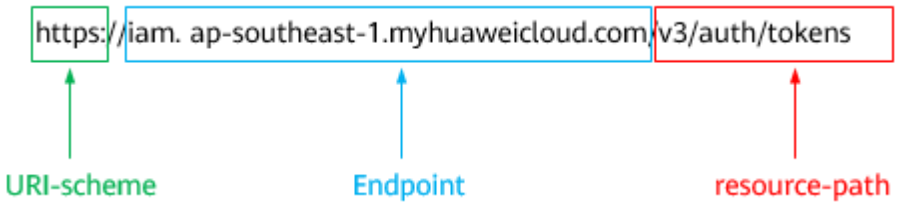
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**. For example, the endpoint of IAM in region **CN-Hong Kong** is **iam.ap-southeast-1.myhuaweicloud.com**.
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.

Parameter	Description
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of *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 **CN-Hong Kong** region, obtain the endpoint of IAM (iam.ap-southeast-1.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:

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Figure 3-1 Example URI



NOTE

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

Request Methods

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

Table 3-2 HTTP methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.

Method	Description
PATCH	Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

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

```
POST https://iam.ap-southeast-1.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.

Table 3-3 Common request header fields

Parameter	Description	Mandatory	Example Value
Host	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of *Hostname:Port number*. If the port number is not specified, the default port is used. The default port number for **https** is **443**.	No This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443
Content-Type	Specifies the type (or format) of the message body. The default value **application/json** is recommended. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495

Parameter	Description	Mandatory	Example Value
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in **Obtaining a Project ID**.	No This field is mandatory for requests that use AK/SK authentication in the Dedicated Cloud (DeC) scenario or multi-project scenario.	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	Specifies the user token. It is a response to the API for **obtaining a user token** (This is the only API that does not require authentication). After the request is processed, the value of **X-Subject-Token** in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZIhvcNAQcCo...ggg1BBIINPXsidG9rZ

NOTE

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

For more details, see "Authentication Using AK/SK" in **Authentication**.

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

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

(Optional) Request Body

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

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

In the case of the API used to **obtain a user token**, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace *username*,

domainname, ******** (login password), and *xxxxxxxxxxxxxxxxx* (project name) with the actual values. Obtain a project name from **Regions and Endpoints**.

NOTE

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

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json

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

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

3.2 Authentication

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

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

Token Authentication

NOTE

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

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

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

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

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

```
POST https://iam.ap-southeast-1.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: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key, which is used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

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 **API Request Signing Guide**.

NOTE

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

3.3 Response

Status Code

After sending a request, you will receive a response, including 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 **Error Codes**.

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

Response Header

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

Figure 3-2 shows the response header fields for the API used to **obtain a user token**. The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

Figure 3-2 Header fields of the response to the request for obtaining a user token

(Optional) Response Body

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

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

```
{
    "token": {
```

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

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

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

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

4 API Description

4.1 Obtaining Details of a DSS Storage Pool

Function

This API is used to obtain the details of a specified DSS storage pool.

URI

GET /v1/{project_id}/pools/{dss_id}

Table 4-1 describes the parameters.

Table 4-1 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID. For how to obtain the project ID, see **Obtaining a Project ID**.
dss_id	Yes	String	The storage pool ID.

Table 4-2 Request parameter

Parameter	Mandatory	Type	Description
usage	No	Boolean	Whether the storage pool capacity information is returned. The value can be **true** or **false**. Value **true** indicates to return the capacity information. The default value is **false**.

Request

Example request

GET https://{endpoint}/v1/{project_id}/pools/{dss_id}?usage=true

Response

Response parameters

Table 4-3 describes the response parameters.

Table 4-3 Response parameters

Parameter	Type	Description
name	String	The storage pool name.
id	String	The storage pool ID.
project_id	String	The ID of the project that the pool belongs.
capacity	Integer	The requested storage pool capacity, in GB.
type	String	The storage pool type. The value can be as follows: • **SAS**: high I/O storage pool • **SSD**: ultra-high I/O storage pool
status	String	The storage pool status. For details, see **DSS Storage Pool Status**.
availability_zone	String	The AZ where the storage pool resides.
created_at	String	The time when the storage pool was created. Time format: UTC YYYY-MM-DDTHH:MM:SS
total_capacity_gb	Integer	The total capacity of the storage pool, in GB. (This parameter is not returned if the storage pool status is **deploying**.)
used_capacity_gb	Integer	The used capacity of the storage pool, in GB. (This parameter is not returned if the storage pool status is **deploying**.)
provisioned_capacity_gb	Integer	The allocated capacity of the storage pool, in GB. (This parameter is not returned if the storage pool status is **deploying**.)

Parameter	Type	Description
max_over_subscription_ratio	Float	The overcommitment ratio of the storage pool. (This parameter is not returned if the storage pool status is **deploying**.)

- Example response

```
{
  "name" : "dedicatedStorage01",
  "id" : "c950ee97-587c-4f24-8a74-3367e3da570f",
  "project_id" : "63d910f2705a487ebe4e1c274748d9e1",
  "capacity" : 100,
  "type" : "SSD",
  "availability_zone" : "AZ1",
  "status" : "available",
  "created_at" : "2014-12-18T15:57:56.299000",
  "total_capacity_gb": 1000,
  "used_capacity_gb": 300,
  "provisioned_capacity_gb":700,
  "max_over_subscription_ratio": 1.0
}
```

- Error response

```
{
 "error": {
   "message": "invalid dss id!",
   "code": "DSS.1001"
 }
}
```

Returned Value

- Normal

  200

- Abnormal

  See **Error Code Description**.

Error Codes

See **Error Codes**.

4.2 Obtaining Details of DSS Storage Pools

Function

This API is used to obtain the DSS storage pools requested by a tenant. Filter query and pagination query are supported.

URI

GET /v1/{project_id}/pools/detail

Table 4-4 describes the parameters.

Table 4-4 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID. For how to obtain the project ID, see **Obtaining a Project ID**.

Table 4-5 Request parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	The maximum number of query results that can be returned. The value must be an integer greater than 0.
offset	No	Integer	The start position of a pagination query. The value must be a positive integer or 0. **NOTE** This parameter indicates that all storage pools after this pagination query offset are queried.
status	No	String	The storage pool status. The value can be **available**, **deploying**, or **extending**. For details, see **DSS Storage Pool Status**.
name	No	String	The storage pool name.
usage	No	Boolean	Whether the storage pool capacity information is returned. The value can be **true** or **false**. Value **true** indicates to return the capacity information. The default value is **false**.

Request

Example request

GET https://{endpoint}/v1/{project_id}/pools/detail?status=available&usage=true

Response

Response parameters

Table 4-6 describes the response parameters.

Table 4-6 Response parameters

Parameter	Type	Description
pools	Array of objects	The storage pool details. For details, see **Table 4-7**.
count	Integer	The number of storage pools.

Description of returned storage pool parameters

Table 4-7 Parameter description

Parameter	Type	Description
name	String	The storage pool name.
id	String	The storage pool ID.
project_id	String	The ID of the project that the pool belongs.
capacity	Integer	The requested storage pool capacity, in TB.
type	String	The storage pool type. The value can be as follows: • **SAS**: high I/O storage pool • **SSD**: ultra-high I/O storage pool
status	String	The storage pool status. For details, see **DSS Storage Pool Status**.
availability_zone	String	The AZ where the storage pool resides.
created_at	String	The time when the storage pool was created. Time format: UTC YYYY-MM-DDTHH:MM:SS
total_capacity_gb	Integer	The total capacity of the storage pool, in GB. (This parameter is not returned if the storage pool status is **deploying**.)
used_capacity_gb	Integer	The used capacity of the storage pool, in GB. (This parameter is not returned if the storage pool status is **deploying**.)

Parameter	Type	Description
provisioned_capacity_gb	Integer	The allocated capacity of the storage pool, in GB. (This parameter is not returned if the storage pool status is **deploying**.)
max_over_subscription_ratio	Float	The overcommitment ratio of the storage pool. (This parameter is not returned if the storage pool status is **deploying**.)

- Example response

```
{
    "pools" : [ {
    "name" : "dedicatedStorage01",
    "id" : "c950ee97-587c-4f24-8a74-3367e3da570f",
    "project_id" : "63d910f2705a487ebe4e1c274748d9e1",
    "capacity" : 1000,
    "type" : "SSD",
    "availability_zone" : "AZ1",
    "status" : "available",
    "created_at" : "2014-12-18T15:57:56.299000",
    "total_capacity_gb": 850,
    "used_capacity_gb": 300,
    "provisioned_capacity_gb":700,
    "max_over_subscription_ratio": 1.0
    },
{
    "name" : "dedicatedStorage02",
    "id" : "6edbc2f4-1507-44f8-ac0d-eed1d2608d38",
    "project_id" : "63d910f2705a487ebe4e1c274748d9e1",
    "capacity" : 1000,
    "type" : "SSD",
    "availability_zone" : "AZ1",
    "status" : "available",
    "created_at" : "2014-12-18T15:57:56.299000",
    "total_capacity_gb": 850,
    "used_capacity_gb": 300,
    "provisioned_capacity_gb":700,
    "max_over_subscription_ratio": 1.0
 } ] ,
   "count": 2
}
```

- Error response

```
{
  "error": {
    "message": "invalid filter limit!",
    "code": "DSS.1003"
  }
}
```

Returned Value

- Normal

  200

- Abnormal

  See **Error Code Description**.

Error Codes

See **Error Codes**.

4.3 Querying Details About All Disks

For details, see **Querying Details About All Disks**.

4.4 Obtaining Details of API Versions

Function

This API is used to query the details of DSS API versions.

URI

GET /

Request

- Request parameters
  None
- Example request
  The following example shows how to query all versions of an API.
  ```
  GET https://{endpoint}/
  ```

Response

Response parameters

Table 4-8 describes the response parameters.

Table 4-8 Response parameters

Parameter	Type	Description
versions	Array	Specifies the API version information.
id	String	Specifies the version ID, for example, **v1**.
links	Array of objects	Specifies the API URL. For details, see **Table 4-9**.
version	String	Specifies the maximum microversion supported by this API.

Parameter	Type	Description
status	String	Specifies the version status. The value can be as follows: **CURRENT**: indicates that the version is currently recommended for use. **SUPPORTED**: indicates that the version is an old version, but is still supported. **DEPRECATED**: indicates a deprecated version which may be deleted later.
updated	String	Specifies the time when the API version was released. Time format: UTC YYYY-MM-DDTHH:MM:SS
min_version	String	Specifies the minimum microversion supported by this API.

Parameters in the **links** field

Table 4-9 describes the parameters in the **links** field.

Table 4-9 Parameter description

Parameter	Type	Description
rel	String	Specifies the link description.
href	String	Specifies the version query link.

- Example response

```
{
    "versions": [
        {
            "min_version": "",
             "links": [
                {
                    "rel": "self",
                    "href": "https://dss.localdomain.com/v1"
                }
            ],
            "id": "v1",
            "updated": "2014-06-28T12:20:21Z",
            "version": "",
            "status": "SUPPORTED"
        },
        {
            "min_version": "",
            "links": [
                {
                    "rel": "self",
                    "href": "https://dss.localdomain.com/v2"
                }
            ],
            "id": "v2",
```

```
            "updated": "2014-06-28T12:20:21Z",
            "version": "",
            "status": "CURRENT"
        }
    ]
}
```

Returned Value

- Normal
  200
- Abnormal
  See **Error Code Description**.

Error Codes

See **Error Codes**.

4.5 Obtaining Details of an API Version

Function

This API is used to query the details of a DSS API version.

URI

GET /{api_version}

Table 4-10 describes the parameter.

Table 4-10 Parameter description

Parameter	Mandatory	Type	Description
api_version	Yes	String	Specifies the target version number. For how to obtain the version number, see **Obtaining Details of API Versions**.

Request

- Request parameters
  None
- Example request
  The following example shows how to query version information of a v1 API.

```
GET https://{endpoint}/v1
```

Response

Response parameters

Table 4-11 describes the response parameters.

Table 4-11 Response parameters

Parameter	Type	Description
version	Object	Specifies the API version information.
id	String	Specifies the version ID, for example, **v1**.
links	Array	Specifies the API URL. For details, see Table 4-12.
version	String	Specifies the maximum microversion supported by this API.
status	String	Specifies the version status. The value can be as follows: **CURRENT**: indicates that the version is currently recommended for use. **SUPPORTED**: indicates that the version is an old version, but is still supported. **DEPRECATED**: indicates a deprecated version which may be deleted later.
updated	String	Specifies the time when the API version was released. Time format: UTC YYYY-MM-DDTHH:MM:SS
min_version	String	Specifies the minimum microversion supported by this API.

Parameters in the **links** field

Table 4-12 describes the parameters in the **links** field.

Table 4-12 Parameter description

Parameter	Type	Description
rel	String	Specifies the link description.
href	String	Specifies the version query link.

- Example response

```
{
   "version": {
          "min_version": "",
           "links": [
             {
                 "rel": "self",
                 "href": "https://dss.localdomain.com/v1"
             }
```

```
            ],
            "id": "v1",
            "updated": "2014-06-28T12:20:21Z",
            "version": "",
            "status": "CURRENT"
        }
}
```

Returned Value

- Normal

  200
- Abnormal

  See **Error Code Description**.

Error Codes

See **Error Codes**.

5 Permissions Policies and Supported Actions

5.1 Introduction

This chapter describes fine-grained permissions management for your DSS resources. If your Huawei Cloud account does not require individual IAM users, you can skip this chapter.

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

You can grant users permissions by using **roles** and **policies**. Roles are a type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Policies define API-based permissions for operations on specific resources under certain conditions, allowing for more fine-grained, secure access control of cloud resources.

NOTE

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

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

Supported Actions

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

- Permission: A statement in a policy that allows or denies certain operations.
- APIs: REST APIs that can be called by a user who has been granted specific permissions.
- Action: Specific operations that are allowed or denied.
- Related actions: Actions on which a specific action depends to take effect. When assigning permissions for the action to a user, you also need to assign permissions for the dependent actions.
- IAM projects or 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 take effect for both IAM and Enterprise Management. Policies that only contain actions for IAM projects can be used and only take effect for IAM. For details about the differences between IAM and enterprise management, see **What Are the Differences Between IAM and Enterprise Management?**

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

- **Storage pool management** actions, including actions supported by DSS storage pool management APIs, such as the APIs for querying a DSS storage pool and querying DSS storage pools.
- **Disk management** actions, including actions supported by DSS disk management APIs, such as the API for querying details about all disks.

5.2 DSS Storage Pool Management

Permissions	APIs	Actions	IAM Project	Enterprise Project
Obtaining details of a DSS storage pool	GET /v1/{project_id}/pools/{dss_id}	dss.action.querypool	√	√
Obtaining details of DSS storage pools	GET /v1/{project_id}/pools/detail	dss.action.listpools	√	√

5.3 Disk Management

Permissions	APIs	Action	IAM Project	Enterprise Project
Querying Details About All Disks by Service	GET /v2/{project_id}/cloudvolumes/detail	evs:volumes:list	√	√

A Appendixes

A.1 DSS Storage Pool Status

Table A-1 Storage pool status

Status	Description
available	The storage pool is available for use.
deploying	The storage pool is being deployed and cannot be used.
extending	The storage pool capacity is being expanded, and the storage pool can be used.

A.2 Error Codes

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in **API Gateway Error Codes**.

Error Code Description

Status Code	Error Code	Error Message	Description	Handling Measure
400	DSS.0001	invalid project id!	Incorrect tenant ID in the URI.	Use the correct project ID.
400	DSS.0002	invalid token!	Header parameters in the HTTP request are incorrect.	Use the correct token.

Status Code	Error Code	Error Message	Description	Handling Measure
400	DSS.0003	invalid token roles!	The token used is incorrect.	Check whether the token has the desired role. If not, add the role for the token.
400	DSS.1001	invalid dss id!	Invalid storage pool ID.	Modify the storage pool ID format.
400	DSS.1002	invalid dss name!	Parameter **name** in the URL for querying the storage pool details is incorrect.	Modify the storage pool name format.
400	DSS.1003	invalid filter limit!	Parameter **limit** in the URL for querying the storage pool details is incorrect.	Enter a value larger than 0 for parameter **limit**.
400	DSS.1004	invalid filter offset!	Parameter **offset** in the URL for querying the storage pool details is incorrect.	Check whether the **offset** parameter in the request is correct.
400	DSS.1005	invalid filter usage!	Parameter **usage** in the URL for querying the storage pool details is incorrect.	Check whether the **usage** parameter in the request is correct.
400	DSS.1006	api roles is null or empty!	User permission error.	Add the required user permission.
400	DSS.1007	User role is not allowed for this action!	You have no permission to the operation.	Add the required user permission.
400	DSS.1008	Type conversion error , parameter type is unexpected	Type conversion error. The parameter type is unexpected.	Check whether the input parameters are correct.
400	DSS.1009	url encoding failed!	Type conversion error.	Check whether the input parameters are correct.
500	DSS.1010	internal error!	The service is unavailable.	Contact technical support.

A.3 Status Codes

- Normal

Status Code	Description
200	OK
201	Created
202	Accepted
204	No Content

- Abnormal

Status Code	Description
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
413	overLimit
415	badMediaType
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout

A.4 Obtaining a Project ID

Scenarios

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

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

Obtain the Project ID by Calling an API

You can obtain a project ID by calling the API used to **query projects based on specified criteria**.

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

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

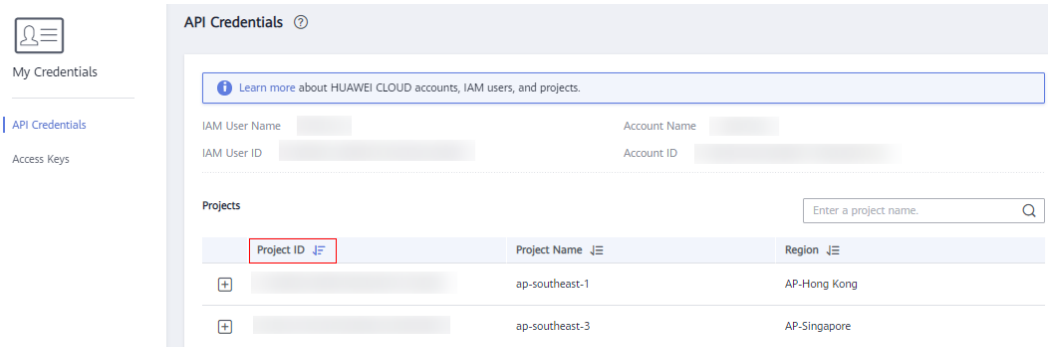
```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "project_name",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
            },
            "id": "a4a5d4098fb4474fa22cd05f897d6b99",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

Obtain a Project ID from the Console

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

1. Log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list.
   On the **API Credentials** page, view the project ID in the project list.

Figure A-1 Viewing the project ID



B Change History

Release On	Description
2018-04-30	This issue is the first official release.

