

Scalable File Service

API Reference (ME-Abu Dhabi Region)

Issue 01
Date 2020-11-06



Copyright © Huawei Technologies Co., Ltd. 2020. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Contents

| | |
|--|-----------|
| 1 Before You Start..... | 1 |
| 1.1 Overview..... | 1 |
| 1.2 API Calling..... | 1 |
| 1.3 Endpoints..... | 1 |
| 1.4 Constraints..... | 2 |
| 1.5 Concepts..... | 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 Getting Started..... | 13 |
| 5 SFS Turbo APIs..... | 15 |
| 5.1 Life Cycle Management..... | 15 |
| 5.1.1 Creating a File System..... | 15 |
| 5.1.2 Deleting a File System..... | 20 |
| 5.1.3 Querying Details About All File Systems..... | 20 |
| 5.1.4 Querying Details About a Single File System..... | 24 |
| 5.2 Storage Capacity Management..... | 26 |
| 5.2.1 Expanding the Capacity of a File System..... | 26 |
| 6 Common Parameters..... | 29 |
| 6.1 SFS Turbo File System Statuses..... | 29 |
| 6.2 SFS Turbo File System Substatuses..... | 29 |
| 7 Appendix..... | 30 |
| 7.1 Status Codes..... | 30 |
| 7.2 Obtaining a Project ID..... | 31 |
| A Change History..... | 33 |

1 Before You Start

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

1.1 Overview

Welcome to *Scalable File Service API Reference*. Scalable File Service (SFS) is a network attached storage (NAS) service that provides scalable, high-performance file storage. With SFS, you can enjoy shared file access spanning multiple Elastic Cloud Servers (ECSs), Bare Metal Servers (BMSs), and containers created on Cloud Container Engine (CCE).

This document describes how to use application programming interfaces (APIs) to perform operations on SFS resources, such as creating, querying, deleting, and updating a file system. For details about all supported operations, see [2 API Overview](#).

If you plan to access SFS through an API, ensure that you are familiar with SFS concepts. For details, see the "Introduction" chapter in the *Scalable File Service User Guide*.

1.2 API Calling

SFS supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS requests. For details about API calling, see [3 Calling APIs](#).

1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoint of SFS, see [Regions and Endpoints](#).

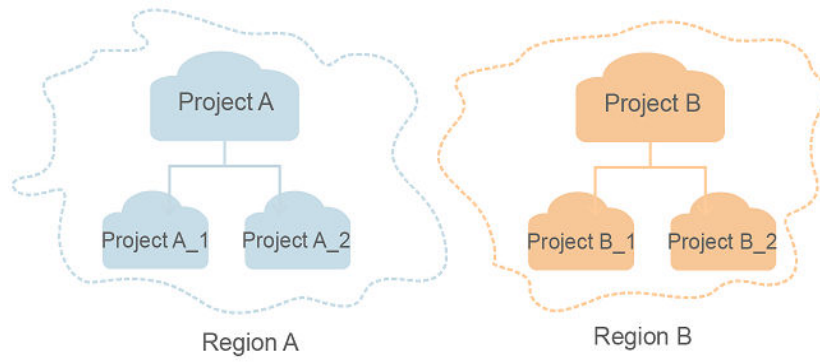
1.4 Constraints

- The numbers of file systems that you can create and their capacities are determined by your quotas. To view or increase the quotas, see the "Quotas" section in the *Scalable File Service User Guide*.
- For more constraints, see API description.

1.5 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. To ensure account security, 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
A region is a geographic area in which cloud resources are deployed. Availability zones (AZs) in the same region can communicate with each other over an intranet, while AZs in different regions are isolated from each other. Deploying cloud resources in different regions can better suit certain user requirements or comply with local laws or regulations.
- AZ
An AZ comprises of one or more physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Computing, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.
- Project
A project corresponds to a region. Default projects are defined to a group and have physically isolated resources (including computing, storage, and network resources) across regions. Users can be granted permissions in a default project to access all resources in the region under their accounts. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

Figure 1-1 Project isolation model



2 API Overview

By using the SFS Turbo APIs, you can create, delete, query SFS Turbo file systems as well as expanding the capacity of an SFS Turbo file system.

Table 2-1 API classification

| Type | Subtype | Description |
|----------------|-----------------------------|---|
| SFS Turbo APIs | Lifecycle management | Include creating file systems, deleting file systems, querying file system lists, and querying file system details. |
| SFS Turbo APIs | Storage capacity management | Expand the capacity of a specified file system. |

3 Calling APIs

3.1 Making an API Request

3.2 Authentication

3.3 Response

3.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for **obtaining a user token** 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

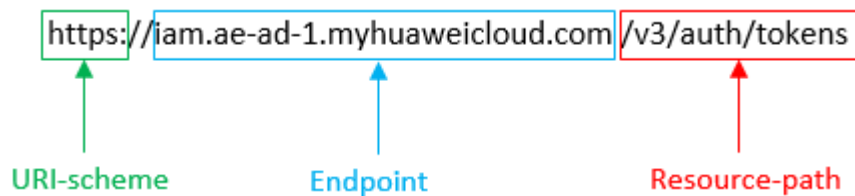
| Parameter | Description |
|---------------|---|
| URI-scheme | Protocol used to transmit requests. All APIs use HTTPS. |
| Endpoint | Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from Regions and Endpoints . For example, the endpoint of IAM in the UAE-Abu Dhabi region is iam.ae-ad-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 " <i>Parameter name=Parameter value</i> ". For example, ?limit=10 indicates that a maximum of 10 data records will be displayed. |

For example, to obtain an IAM token in the **UAE-Abu Dhabi** region, obtain the endpoint of IAM (**iam.ae-ad-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.ae-ad-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.ae-ad-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 <i>Hostname:Port number</i> . 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 7.2 Obtaining a Project ID . | No | e9993fc787d94b6c886cbaa340f9c0f4 |
| X-Auth-Token | Specifies the user token. It is a response to the API for obtaining a user token (only this API 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: MIIPAgYJKoZlhvcNAQcCo...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 [3.2 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.ae-ad-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 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 *xxxxxxxxxxxxxxxxxxxx* (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.ae-ad-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": "xxxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

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

3.2 Authentication

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

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. AK/SK-based authentication is recommended because it is more secure than token-based authentication.

Token-based Authentication

 NOTE

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

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

You can obtain a token by calling the [Obtaining User Token](#) API. When you call the API, 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.ae-ad-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK-based Authentication

NOTE

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

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

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

In AK/SK-based authentication, you can use an AK/SK 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 [7.1 Status Codes](#).

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

Response Header

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

[Figure 3-2](#) 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

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token
→ MIIVXQYJKoZIhvcNAQcCoIIYJCCEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOansiZXhwaXJlc19hdCI6ijlwMTktMDItMTNUMC
fj3KIs6YgKnpVNRbW2eZ5eb78SZ0kqjACgkIQ1wi4JIGzrpd18LGXK5tdfq4lqHCYb8P4NaY0NyejcAgzJVeFYtLWT1GSO0zxKZmlQHQj82HBqHdglZO9fuEbL5dMhdavj+33wEI
xHRC9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXI1jipPEGA270g1FruooL6jqglFkNPQuFSOUB+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUUpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==
x-xss-protection → 1; mode=block;
```

(Optional) Response Body

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

The following is part of the response body for the API used to [obtain a user token](#).

```
{
  "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 Getting Started

This section describes how to use APIs by calling an API to create an SFS Turbo file system.

NOTE

The token obtained from IAM is valid for only 24 hours. If you want to use one token for authentication, you can cache it to avoid frequently calling.

Involved APIs

If you use a token for authentication, you must obtain the token and add **X-Auth-Token** to the request header of the API when making a call. The following APIs are involved in the request for creating an SFS Turbo file system:

- API for obtaining tokens from IAM
- API for creating an SFS Turbo file system. For details, see [5.1.1 Creating a File System](#).

Procedure

Step 1 Obtain the token by following instructions in [3.2 Authentication](#).

Step 2 Add **X-Auth-Token** to the request header.

Step 3 Specify the following parameters in the request body:

```
{
  "share": {
    "name": "sfs-turbo-test",
    "share_proto": "NFS",
    "share_type": "STANDARD",
    "size": 100,
    "availability_zone": "az1",
    "vpc_id": "d651ea2b-2b20-4c6d-8bbf-2adcec18dac9",
    "subnet_id": "b8884abe-f47b-4917-9f6c-f64825c365db",
    "security_group_id": "8c4ebbd0-6edf-4aae-8353-81ce6d06e1f4"
  }
}
```

Step 4 Send the request **POST** https://Endpoint of SFS Turbo/v1/{project_id}/sfs-turbo/shares.

Step 5 After the request is successfully responded, the ID and name of the SFS Turbo file system are returned.

If the request fails, an error code and error information are returned. For details about the error codes, see the abnormal return values of the corresponding API.

Query SFS Turbo file system details based on the returned file system ID. For details, see [5.1.4 Querying Details About a Single File System](#).

If the returned status of the file system is **200**, the SFS Turbo file system is successfully created. For details about the return values of request exceptions, see the abnormal return values of the corresponding API. For other statuses, see [6.1 SFS Turbo File System Statuses](#).

You can query and delete an SFS Turbo file system based on the file system ID.

----End

Configuration Example

If the token has been obtained, you can run the following **curl** command to create an SFS Turbo file system:

```
curl -k -i -X POST -H "X-Auth-Token: token_value" -H "Content-Type: application/json" -d '{"share": {"name": "sfs-turbo-test", "share_proto": "NFS", "share_type": "STANDARD", "size": 100, "availability_zone": "az1", "vpc_id": "d651ea2b-2b20-4c6d-8bbf-2adcec18dac9", "subnet_id": "b8884abe-f47b-4917-9f6c-f64825c365db", "security_group_id": "8c4ebbd0-6edf-4aae-8353-81ce6d06e1f4"}}' "https://127.0.0.1:8979/v1/xxx/bx/bex5cfx41f0a08ay915fd79240d/sfs-turbo/shares"
```

5 SFS Turbo APIs

[5.1 Life Cycle Management](#)

[5.2 Storage Capacity Management](#)

5.1 Life Cycle Management

5.1.1 Creating a File System

Function

This API is used to create an SFS Turbo file system.

URI

- URI format
POST /v1/{project_id}/sfs-turbo/shares
- Parameter description

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| project_id | Yes | String | Specifies the project ID. For details about how to obtain the project ID, see 7.2 Obtaining a Project ID . |

Request

- Parameter description

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--|
| share | Yes | Object | Specifies information about an SFS Turbo file system. For details about the parameters, see the description of the share field . |

- Description of the **share** field

| Parameter | Mandatory | Type | Description |
|----------------|-----------|--------|---|
| name | Yes | String | Specifies the name of an SFS Turbo file system. The value contains 4 to 64 characters and must start with a letter. This value can contain letters (case insensitive), digits, hyphens (-), and underscores (_), and cannot contain other special characters. |
| share_protocol | No | String | Specifies the protocol for sharing file systems. The valid value is NFS . Network File System (NFS) is a distributed file system protocol that allows different computers and operating systems to share data over a network. |

| Parameter | Mandatory | Type | Description |
|-------------------|-----------|---------|---|
| share_type | Yes | String | Specifies the file system type. The valid values are STANDARD and PERFORMANCE . STANDARD : Standard file system, corresponding to the media of SAS disks. PERFORMANCE : Performance file system, corresponding to the media of SSD disks. |
| size | Yes | Integer | Specifies the capacity of a common file system, in GB. The value ranges from 500 to 32768. (OBT users can create 40 GB or larger file systems.) |
| availability_zone | Yes | String | Specifies the code of the AZ where the file system is located. For details about the code, see Regions and Endpoints . |
| vpc_id | Yes | String | Specifies the VPC ID of a tenant in a region. You can obtain the VPC ID from the console or by following the instructions provided in "Querying VPCs" in . |

| Parameter | Mandatory | Type | Description |
|-----------------------|-----------|--------|--|
| subnet_id | Yes | String | Specifies the network ID of the subnet of a tenant in a VPC. You can obtain the network ID from the VPC console or by following the instructions provided in "Querying Subnets" in <i>Virtual Private Cloud API Reference</i> . |
| security_group_id | Yes | String | Specifies the security group ID of a tenant in a region. You can obtain the security group ID from the console or by following the instructions provided in "Querying Security Groups" in <i>Virtual Private Cloud API Reference</i> . |
| enterprise_project_id | No | String | Specifies the ID of the enterprise project bound to the file system when a file system is created. This is not supported by the current version. |
| backup_id | No | String | Specifies the backup ID. This parameter is mandatory when you create a file system from a backup. This is not supported by the current version. |
| description | No | String | Specifies the file system description. The length is 0-255 characters. This is not supported by the current version. |

 NOTE

- The regions mentioned above are the same region. Currently, cross-region configuration is not supported.
- SFS Turbo will create two private IP addresses and one virtual IP address under the subnet you specified.
- To ensure normal use, SFS Turbo will enable the inbound rules for ports **111**, **445**, **2049**, **2051**, **2052**, and **20048** in the security group you specified.
- An ECS cannot access file systems on VPCs other than the one where the ECS resides. Make sure that you enter the ID of the VPC when creating a file system to be the VPC where the ECS resides for mounting the file system.

- Example request

```
{
  "share": {
    "name": "sfs-turbo-test",
    "share_proto": "NFS",
    "share_type": "STANDARD",
    "size": 100,
    "availability_zone": "az1",
    "vpc_id": "d651ea2b-2b20-4c6d-8bbf-2adcec18dac9",
    "subnet_id": "b8884abe-f47b-4917-9f6c-f64825c365db",
    "security_group_id": "8c4ebbd0-6edf-4aae-8353-81ce6d06e1f4"
  }
}
```

Response

- Parameter description

| Parameter | Type | Description |
|-----------|--------|---|
| id | String | Specifies the ID of an SFS Turbo file system. |
| name | String | Specifies the name of an SFS Turbo file system. |
| status | String | Specifies the status of an SFS Turbo file system. For details, see 6.1 SFS Turbo File System Statuses . |

- Example response

```
{
  "id": "708c017c-54b5-429a-a098-7692e23fa518",
  "name": "sfs-turbo-test",
  "status": "100"
}
```

Status Codes

- Normal

202

- Abnormal

For details, see [7.1 Status Codes](#).

5.1.2 Deleting a File System

Function

This API is used to delete an SFS Turbo file system.

URI

- URI format
DELETE /v1/{project_id}/sfs-turbo/shares/{share_id}
- Parameter description

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| project_id | Yes | String | Specifies the project ID. For details about how to obtain the project ID, see 7.2 Obtaining a Project ID . |
| share_id | Yes | String | Specifies the ID of an SFS Turbo file system. |

Request

- None

Response

- None

Status Codes

- Normal
202
- Abnormal

For details, see [7.1 Status Codes](#).

5.1.3 Querying Details About All File Systems

Function

This API is used to query details about all SFS Turbo file systems.

URI

- URI format
GET /v1/{project_id}/sfs-turbo/shares/detail?limit={limit}&offset={offset}

- Parameter description

| Parameter | Mandatory | Type | Description |
|------------|----------------------|---------|--|
| project_id | Yes | String | Specifies the project ID. For details about how to obtain the project ID, see 7.2 Obtaining a Project ID . |
| limit | No (query parameter) | Integer | Specifies the number of returned file systems. The maximum value is 200 . |
| offset | No (query parameter) | Integer | Specifies the offset of the number of queried file systems. |

Request

- Parameter description
None
- Example request
None

Response

- Parameter description

| Parameter | Type | Description |
|-----------|------------------|---|
| shares | Array of objects | Specifies the list of SFS Turbo file systems. For details, see the description of the share field . |

- Description of the **share** field

| Parameter | Type | Description |
|-----------|--------|---|
| id | String | Specifies the ID of an SFS Turbo file system. |
| name | String | Specifies the name of the SFS Turbo file system. |
| status | String | Specifies the SFS Turbo file system status. For details, see 6.1 SFS Turbo File System Statuses . |

| Parameter | Type | Description |
|-------------------|------------------|--|
| sub_status | String | Specifies the SFS Turbo sub-status. For details, see 6.2 SFS Turbo File System Substatuses . |
| version | String | Specifies the version ID of an SFS Turbo file system. |
| created_at | String | Specifies the creation time. UTC time, for example: 2018-11-19T04:02:03 |
| export_location | String | Specifies the mount point of the SFS Turbo file system. |
| action_progress | Object | Specifies the creation progress of the SFS Turbo file system. |
| actions | Array of strings | Specifies the action of the SFS Turbo file system. Currently, this parameter is reserved. |
| share_type | String | Specifies the type of the SFS Turbo file system. The value can be STANDARD or PERFORMANCE . |
| region | String | Specifies the region of the SFS Turbo file system. |
| availability_zone | String | Specifies the code of the AZ where the SFS Turbo file system is located. |
| az_name | String | Specifies the name of the AZ where the SFS Turbo file system is located. |
| vpc_id | String | Specifies the VPC ID specified by the user. |
| subnet_id | String | Specifies the network ID of the subnet specified by the user. |
| security_group_id | String | Specifies the ID of a security group specified by the user. |
| size | String | Specifies the total capacity of the SFS Turbo file system in the unit of GB. |
| avail_capacity | String | Specifies the available capacity of the SFS Turbo file system in the unit of GB. |

| Parameter | Type | Description |
|-------------|--------|--|
| share_proto | String | Specifies the protocol type of the SFS Turbo file system. The current value is NFS . |
| expand_type | String | For an enhanced file system, bandwidth is returned for this field. Otherwise, bandwidth is not returned. |

- Example response

```
{
  "shares": [
    {
      "id": "8fba8253-c914-439d-ae8b-d5c89d0bf5e8",
      "name": "sfs-turbo-8468",
      "status": "200",
      "version": "1.0.0",
      "actions": [],
      "region": "north-1",
      "created_at": "2018-11-19T04:02:03",
      "export_location": "192.168.0.90:/",
      "action_progress": {},
      "share_type": "STANDARD",
      "sub_status": "330",
      "availability_zone": "az1.dc1",
      "az_name": "AZ 1",
      "vpc_id": "b24e39e1-bc0c-475b-ae0c-aef9cf240af3",
      "subnet_id": "86fc01ea-8ec8-409d-ba7a-e0ea16d4fd97",
      "security_group_id": "50586458-aec9-442c-bb13-e08ddc6f1b7a",
      "size": "0.0",
      "avail_capacity": "0.0",
      "share_proto": "NFS",
      "expand_type": "bandwidth"
    },
    {
      "id": "65f2d30b-7b4e-4786-9608-4324faef6646",
      "name": "sfs-turbo-df12",
      "status": "200",
      "version": "1.0.0",
      "actions": [],
      "region": "north-1",
      "created_at": "2018-11-15T02:32:10",
      "export_location": "192.168.0.197:/",
      "action_progress": {},
      "share_type": "STANDARD",
      "availability_zone": "az1.dc1",
      "az_name": "AZ 1",
      "vpc_id": "b24e39e1-bc0c-475b-ae0c-aef9cf240af3",
      "subnet_id": "86fc01ea-8ec8-409d-ba7a-e0ea16d4fd97",
      "security_group_id": "50586458-aec9-442c-bb13-e08ddc6f1b7a",
      "size": "0.0",
      "avail_capacity": "0.0",
      "share_proto": "NFS"
    }
  ]
}
```

Status Codes

- Normal
200

- Abnormal

For details, see [7.1 Status Codes](#).

5.1.4 Querying Details About a Single File System

Function

This API is used to query details about an SFS Turbo file system.

URI

- URI format
GET /v1/{project_id}/sfs-turbo/shares/{share_id}
- Parameter description

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| project_id | Yes | String | Specifies the project ID. For details about how to obtain the project ID, see 7.2 Obtaining a Project ID . |
| share_id | Yes | String | Specifies the UUID of an SFS Turbo file system. |

Request

- Parameter description
None
- Example request
None

Response

- Parameter description

| Parameter | Type | Description |
|-----------|--------|---|
| id | String | Specifies the ID of an SFS Turbo file system. |
| name | String | Specifies the name of the SFS Turbo file system. |
| status | String | Specifies the SFS Turbo file system status. For details, see 6.1 SFS Turbo File System Statuses . |

| Parameter | Type | Description |
|-------------------|------------------|--|
| sub_status | String | Specifies the SFS Turbo sub-status. For details, see 6.2 SFS Turbo File System Substatuses . |
| version | String | Specifies the version ID of an SFS Turbo file system. |
| created_at | String | Specifies the creation time. UTC time, for example: 2018-11-19T04:02:03 |
| export_location | String | Specifies the mount point of the SFS Turbo file system. |
| action_progress | Object | Specifies the creation progress of the SFS Turbo file system. |
| actions | Array of strings | Specifies the action of the SFS Turbo file system. Currently, this parameter is reserved. |
| share_type | String | Specifies the type of the SFS Turbo file system. The value can be STANDARD or PERFORMANCE . |
| region | String | Specifies the region of the SFS Turbo file system. |
| availability_zone | String | Specifies the code of the AZ where the SFS Turbo file system is located. |
| az_name | String | Specifies the name of the AZ where the SFS Turbo file system is located. |
| vpc_id | String | Specifies the VPC ID specified by the user. |
| subnet_id | String | Specifies the network ID of the subnet specified by the user. |
| security_group_id | String | Specifies the ID of a security group specified by the user. |
| size | String | Specifies the total capacity of the SFS Turbo file system in the unit of GB. |
| avail_capacity | String | Specifies the available capacity of the SFS Turbo file system in the unit of GB. |
| share_proto | String | Specifies the protocol type of the SFS Turbo file system. The current value is NFS . |

| Parameter | Type | Description |
|-------------|--------|--|
| expand_type | String | For an enhanced file system, bandwidth is returned for this field. Otherwise, bandwidth is not returned. |

- Example response

```
{
  "id": "8fba8253-c914-439d-ae8b-d5c89d0bf5e8",
  "name": "sfs-turbo-8468",
  "status": "200",
  "version": "1.0.0",
  "actions": [],
  "region": "north-1",
  "created_at": "2018-11-19T04:02:03",
  "export_location": "192.168.0.90:/",
  "action_progress": {},
  "share_type": "STANDARD",
  "sub_status": "330",
  "availability_zone": "az1.dc1",
  "az_name": "AZ 1",
  "vpc_id": "b24e39e1-bc0c-475b-ae0c-ae9cf240af3",
  "subnet_id": "86fc01ea-8ec8-409d-ba7a-e0ea16d4fd97",
  "security_group_id": "50586458-aec9-442c-bb13-e08ddc6f1b7a",
  "size": "0.0",
  "avail_capacity": "0.0",

  "share_proto": "NFS",
  "expand_type": "bandwidth"
}
```

Status Codes

- Normal

200

- Abnormal

For details, see [7.1 Status Codes](#).

5.2 Storage Capacity Management

5.2.1 Expanding the Capacity of a File System

Function

This API is used to expand the capacity of an SFS Turbo file system. Capacity expansion is an asynchronous operation. You can check whether the expansion is successful by checking field **sub_status** returned by [Querying Details About a Single File System](#). If the value of the sub-status is **221**, the expansion is successful.

URI

- URI format

POST /v1/{project_id}/sfs-turbo/shares/{share_id}/action

- Parameter description

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| project_id | Yes | String | Specifies the project ID. For details about how to obtain the project ID, see 7.2 Obtaining a Project ID . |
| share_id | Yes | String | Specifies the UUID of an SFS Turbo file system. |

Request

- Parameter description

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--|
| extend | Yes | Object | Specifies the extend object. For details, see the parameter in the extend field . |

- Parameter in the **extend** field

| Parameter | Mandatory | Type | Description |
|-----------|-----------|---------|---|
| new_size | Yes | Integer | Specifies the post-expansion capacity (GB) of the shared file system. The capacity expansion step is greater than or equal to 100 GB. For a common file system, the value of capacity ranges from 500 to 32768. |

- Example request

```
{
  "extend": {
    "new_size": 500
  }
}
```

Response

- Parameter description

| Parameter | Type | Description |
|-----------|--------|---|
| id | String | Specifies the ID of an SFS Turbo file system. |
| name | String | Specifies the name of an SFS Turbo file system. |

- Example response

```
{  
  "id": "67d4bd5e-7b2f-4c24-9a0b-c0038940c6f8",  
  "name": "sfs-turbo-cts"  
}
```

Status Codes

- Normal

202

- Abnormal

For details, see [7.1 Status Codes](#).

6 Common Parameters

[6.1 SFS Turbo File System Statuses](#)

[6.2 SFS Turbo File System Substatuses](#)

6.1 SFS Turbo File System Statuses

- SFS Turbo file system status elements

| Returned Value | Description |
|----------------|--|
| 100 | CREATING: The file system is being created. |
| 200 | ACTIVE: The file system is active. An SFS Turbo file system can be mounted in this status. |
| 300 | FAILED: The job failed. |
| 303 | CREATE_FAILED: The cluster failed to be created. |
| 400 | DELETED: The cluster has been deleted. |
| 800 | FROZEN: The cluster has been frozen. |

6.2 SFS Turbo File System Substatuses

- SFS Turbo file system substatus elements

| Returned Value | Description |
|----------------|--|
| 121 | Expanding the capacity online. |
| 221 | Online capacity expansion succeeded. |
| 321 | Failed to perform online capacity expansion. |

7 Appendix

[7.1 Status Codes](#)

[7.2 Obtaining a Project ID](#)

7.1 Status Codes

- Normal

| Returned Value | Description |
|----------------|---|
| 200 OK | Specifies the normal response for the GET and PUT operations. |
| 201 Created | Specifies the normal response for the POST operation. |
| 202 Accepted | The request has been accepted for processing. |
| 204 No Content | Specifies the normal response for the DELETE operation. |

- Abnormal

| Returned Value | Description |
|------------------------|--|
| 400 Bad Request | The server failed to process the request. |
| 401 Unauthorized | You must enter a username and the password to access the requested page. |
| 403 Forbidden | Access to the requested page is forbidden. |
| 404 Not Found | The requested page was not found. |
| 405 Method Not Allowed | You are not allowed to use the method specified in the request. |

| Returned Value | Description |
|-----------------------------------|---|
| 406 Not Acceptable | The response generated by the server could not be accepted by the client. |
| 407 Proxy Authentication Required | You must use the proxy server for authentication. Then the request can be processed. |
| 408 Request Timeout | The request timed out. |
| 409 Conflict | The request could not be processed due to a conflict. |
| 500 Internal Server Error | The request is not completed because of a service error. |
| 501 Not Implemented | The request is not completed because the server does not support the requested function. |
| 502 Bad Gateway | The request is not completed because the server receives an invalid response from an upstream server. |
| 503 Service Unavailable | The request is not completed because the service is unavailable. |
| 504 Gateway Timeout | A gateway timeout error occurred. |

7.2 Obtaining a Project ID

Scenarios

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

- [Obtain the Project ID by Calling an API](#)
- [Obtain the Project ID from the Console](#)

Obtain the Project ID by Calling an API

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

The API used to obtain a project ID is GET `https://{Endpoint}/v3/projects`. {Endpoint} is the IAM endpoint and can be obtained from [Regions and Endpoints](#). For details about API authentication, see [3.2 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 **My Credentials** page, view the project ID in the project list.

A Change History

| Release Date | What's New |
|--------------|---|
| 2020-11-03 | This issue is the first official release. |