

弹性文件服务

## API 参考

文档版本 01  
发布日期 2025-12-17



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目 录

1 使用前必读.....	1
2 API 概览.....	3
3 如何调用 API.....	5
3.1 构造请求.....	5
3.2 认证鉴权.....	7
3.2.1 用户签名验证.....	7
3.2.2 Header 中携带签名.....	9
3.2.3 签名计算工具.....	16
3.3 返回结果.....	16
4 快速入门.....	18
4.1 创建通用文件系统.....	18
4.2 获取通用文件系统列表.....	21
5 API 说明.....	24
5.1 通用文件系统管理.....	24
5.1.1 创建通用文件系统 - CreateFileSystem.....	24
5.1.2 删除通用文件系统 - DeleteFileSystem.....	26
5.1.3 获取通用文件系统列表 - ListFileSystems.....	27
5.2 访问规则管理.....	29
5.2.1 设置通用文件系统 ACL - PutACL.....	29
5.2.2 获取通用文件系统 ACL - GetACL.....	32
5.2.3 删除通用文件系统 ACL - DeleteACL.....	34
5.3 标签管理.....	35
5.3.1 批量添加资源标签 - TagResources.....	35
5.3.2 批量删除资源标签 - UnTagResources.....	38
5.3.3 查询资源标签 - ListTagsForResource.....	40
5.3.4 查询资源实例列表 - ListResourcesByTag.....	42
5.3.5 查询资源实例数量 - ListResourceQuantityByTag.....	46
5.3.6 查询项目标签 - ListTagsByProject.....	49
6 权限和授权项.....	52
6.1 权限及授权项说明.....	52
6.2 授权项分类.....	53

<b>7 附录.....</b>	<b>56</b>
7.1 状态码.....	56
7.2 错误码.....	57
7.3 获取访问密钥（AK/SK） .....	62
7.4 获取项目 ID.....	62

# 1 使用前必读

## 概述

欢迎使用弹性文件服务（ Scalable File Service, SFS ）。弹性文件服务提供按需扩展的高性能文件存储（ Network Attached Storage, NAS ），可为云上多个弹性云服务器（ Elastic Cloud Server, ECS ），容器（ Cloud Container Engine, CCE ），裸金属服务器（ Bare Metal Server, BMS ）提供共享访问。

您可以使用本文档提供API对通用文件系统进行相关操作，如创建、查询、删除等。支持的全部操作请参见[API概览](#)。

在调用弹性文件服务API之前，请确保已经充分了解弹性文件服务相关概念，详细信息请参见《[弹性文件服务产品介绍](#)》。

## 调用说明

弹性文件服务提供了REST（ Representational State Transfer ）风格API，支持您通过HTTP/HTTPS请求调用，调用方法请参见[如何调用API](#)。

## 终端节点

终端节点（ Endpoint ）即调用API的[请求地址](#)，不同服务不同区域的终端节点不同，您可以从下表中查询弹性文件服务的终端节点。

表 1-1 地区和终端节点

区域名称	区域	终端节点（ Endpoint ）	协议类型
华北-北京四	cn-north-4	sfs3.cn-north-4.myhuaweicloud.com	HTTPS
华东-上海一	cn-east-3	sfs3.cn-east-3.myhuaweicloud.com	HTTPS

区域名称	区域	终端节点 ( Endpoint )	协议类型
华南-广州	cn-south-1	sfs3.cn-south-1.myhuaweicloud.com	HTTPS
西南-贵阳一	cn-southwest-2	sfs3.cn-southwest-2.myhuaweicloud.com	HTTPS
中国-香港	ap-southeast-1	sfs3.ap-southeast-1.myhuaweicloud.com	HTTPS

约束限制

- 通用文件系统API只支持内网调用，不支持公网调用。
- 更详细的限制请参见[具体API的说明](#)。

基本概念

- 账号  
用户注册时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用用户进行日常管理工作。
- 用户  
由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。  
通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。
- 区域（ Region ）  
从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region，通用Region指面向公共租户提供通用云服务的Region；专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。  
详情请参见[区域和可用区](#)。
- 可用区（ AZ， Availability Zone ）  
一个可用区是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

# 2 API 概览

通过配合使用弹性文件服务所提供的接口和OpenStack原生接口，您可以完整地使用弹性文件服务的所有功能。

使用企业项目功能时，关于API接口是否支持企业项目，请参考[API授权项列表](#)。

当本文描述内容与上述开源社区的描述不一致时，请以本文描述为准。

## 通用文件系统管理接口

表 2-1 通用文件系统管理接口

接口	说明
<a href="#">创建通用文件系统 - CreateFileSystem</a>	创建一个新的通用文件系统。
<a href="#">删除通用文件系统 - DeleteFileSystem</a>	删除一个指定的通用文件系统。
<a href="#">获取通用文件系统列表 - ListFileSystems</a>	获取通用文件系统的列表。

## 访问规则管理接口

表 2-2 访问规则管理接口

接口	说明
<a href="#">设置通用文件系统ACL - PutACL</a>	设置一个指定通用文件系统的ACL信息。
<a href="#">获取通用文件系统ACL - GetACL</a>	获取一个指定通用文件系统的ACL信息。
<a href="#">删除通用文件系统ACL - DeleteACL</a>	删除一个指定通用文件系统的ACL信息。

标签管理接口

表 2-3 标签管理接口

接口	说明
批量添加资源标签 - TagResources	为指定通用文件系统批量添加标签。
批量删除资源标签 - UnTagResources	为指定通用文件系统批量删除标签。
查询资源标签 - ListTagsForResource	查询指定通用文件系统的标签。
查询资源实例列表 - ListResourcesByTag	使用标签过滤通用文件系统。优先按照通用文件系统创建时间倒序排序。
查询资源实例数量 - ListResourceQuantityByTag	根据标签查询通用文件系统的数量。
查询项目标签 - ListTagsByProject	查询租户在指定Project中所有通用文件系统的标签集合。

# 3 如何调用 API

## 3.1 构造请求

本节介绍REST API请求的组成。

### 请求 URI

SFS根据通用文件系统及携带的资源参数来确定具体的URI，当需要进行资源操作时，可以使用这个URI地址。

URI的一般格式为（方括号内为可选项）：

**protocol://[filesystem.]domain[:port]/[?param]**

表 3-1 URI 中的参数

参数	描述	是否必选
protocol	请求使用的协议类型，如HTTP、HTTPS。HTTPS表示通过安全的HTTPS访问该资源，SFS服务支持HTTP，HTTPS两种传输协议。	必选
filesystem	请求使用的通用文件系统资源路径，在整个系统中唯一标识一个通用文件系统。	可选
domain	存放资源的服务器的域名或IP地址。	必选
port	请求使用的端口号。根据软件服务器的部署不同而不同。缺省时使用默认端口，各种传输协议都有默认的端口号，如HTTP的默认端口为80，HTTPS的默认端口为443。 SFS服务的http方式访问端口为80，HTTPS方式访问端口为443。	可选
param	请求使用的通用文件系统的具体资源，缺省为请求通用文件系统自身资源。	可选

须知

除获取通用文件系统列表之外的所有接口，都应当包含通用文件系统名。SFS基于DNS解析性能和可靠性的考虑，要求凡是携带通用文件系统名的请求，在构造URL的时候都必须将通用文件系统名放在domain前面，形成三级域名形式，又称为虚拟主机访问域名。

请求方法

HTTP方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

表 3-2 SFS 支持的 REST 请求方法

方法	说明
GET	请求服务器返回指定资源，如获取通用文件系统列表等。
PUT	请求服务器更新指定资源，如创建通用文件系统等。
POST	请求服务器新增资源或执行特殊操作。
DELETE	请求服务器删除指定资源，如删除通用文件系统等。
HEAD	请求服务器返回指定资源的概要。
OPTIONS	请求服务器检查是否具有某个资源的操作权限。

请求消息头

可选的附加请求头字段，如指定的URI和HTTP方法所要求的字段。详细的公共请求消息头字段请参见表3-3。

表 3-3 公共请求消息头

消息头名称	描述	是否必选
Authorization	请求消息中可带的签名信息。 类型：字符串。 默认值：无。	必选
Content-Length	RFC 2616中定义的消息（不包含消息头）长度。 类型：字符串。 默认值：无。 条件：PUT操作和加载XML的操作必须带。	有条件必选
Content-Type	资源内容的类型，例如：text/plain。 类型：字符串。 默认值：无。	否

消息头名称	描述	是否必选
Date	请求发起端的日期和时间，例如：Wed, 27 Jun 2018 13:39:15 +0000。 类型：字符串。 默认值：无。 条件：如果是消息头中带了x-obs-date字段，则可以不带该字段，其他情况下必选。	有条件必选
Host	表明主机地址。如 filesystem.sfs3.region.myhuaweicloud.com。 类型：字符串。 默认值：无。	是

## 请求消息体（可选）

请求消息体通常以结构化格式（如JSON或XML）发出，与请求消息头中Content-type对应，传递除请求消息头之外的内容。若请求消息体中参数支持中文，则中文字符必须为UTF-8编码。

每个接口的请求消息体内容不同，也并不是每个接口都需要有请求消息体（或者说消息体为空），GET、DELETE操作类型的接口就不需要消息体，消息体具体内容需要根据具体接口而定。

## 发起请求

共有两种方式可以基于已构建好的请求消息发起请求，分别为：

- cURL  
cURL是一个命令行工具，用来执行各种URL操作和信息传输。cURL充当的是HTTP客户端，可以发送HTTP请求给服务端，并接收响应消息。cURL适用于接口调试。关于cURL详细信息请参见<https://curl.haxx.se/>。由于cURL无法计算签名，使用cURL时仅支持访问匿名的公共SFS资源。
- 编码  
通过编码调用接口，组装请求消息，并发送处理请求消息。可以自行编码实现。

## 3.2 认证鉴权

### 3.2.1 用户签名验证

SFS通过AK/SK对请求进行签名，在向SFS发送请求时，客户端发送的每个消息头需要包含由SK、请求时间、请求类型等信息生成的签名信息。

- AK(Access Key ID)：访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和私有访问密钥一起使用，对请求进行加密签名。格式例如：  
HCY8BGCN1YM5ZWYOK1MH

- SK(Secret Access Key)：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。格式例如：  
9zYwf1uabSQY0JTnFqbUqG7vcfqYBaTdXde2GUcq

用户可以在IAM服务中获取AK和SK，获取的方法请参见[获取访问密钥（AK/SK）](#)。

SFS根据应用场景，提供了[Header中携带签名](#)的签名计算方式。

以Header中携带签名为例，用户签名验证流程如[表3-4](#)所示。Header中携带签名方法的具体参数说明及代码示例，请参见[Header中携带签名](#)。

表 3-4 SFS 签名计算和验证步骤

步骤		示例
签名计算	1. 构造HTTP消息	PUT /HTTP/1.1 Host: filesystem.sfs3.region.myhuaweicloud.com Date: Tue, 04 Jun 2019 06:54:59 GMT Content-Type: text/plain Content-Length: 5913
	2. 按照签名规则计算StringToSign	StringToSign = HTTP-Verb + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Date + "\n" + CanonicalizedHeaders + CanonicalizedResource
	3. 准备AK和SK	AK: ***** SK: *****
	4. 计算签名Signature	Signature = Base64( HMAC-SHA1( <b>SecretAccessKeyID</b> , UTF-8-Encoding-Of( <b>StringToSign</b> ) ) )
	5. 添加签名头域发送到SFS服务	PUT /object HTTP/1.1 Host: filesystem.sfs3.region.myhuaweicloud.com Date: Tue, 04 Jun 2019 06:54:59 GMT Content-Type: text/plain Content-Length: 5913 Authorization: OBS <b>AccessKeyID:Signature</b>
签名验证	6. 接收HTTP消息	PUT / HTTP/1.1 Host: filesystem.sfs3.region.myhuaweicloud.com Date: Tue, 04 Jun 2019 06:54:59 GMT Content-Type: text/plain Content-Length: 5913 Authorization: OBS <b>AccessKeyID:Signature</b>
	7. 根据请求中的AK获取SK	从头域Authorization中取出AK，去IAM取回用户的SK

步骤	示例
8. 按照签名规则计算 <b>StringToSign</b>	StringToSign = HTTP-Verb + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Date + "\n" + CanonicalizedHeaders + CanonicalizedResource
9. 计算签名 <b>Signature</b>	Signature = Base64( HMAC-SHA1( <b>SecretAccessKeyID</b> , UTF-8-Encoding-Of( <b>StringToSign</b> ) ) )
10. 验证签名	验证头域Authorization中的 <b>Signature</b> 与服务端计算的 <b>Signature</b> 是否相等 相等：签名验证通过 不相等：签名验证失败

### 3.2.2 Header 中携带签名

通用文件系统的所有API接口都可以通过在header中携带签名方式来进行身份认证。

在Header中携带签名是指将通过HTTP消息中Authorization header头域携带签名信息，消息头域的格式为：

```
Authorization: OBS AccessKeyID:signature
```

签名的计算过程如下：

- 1、构造请求字符串(StringToSign)。
- 2、对第一步的结果进行UTF-8编码。
- 3、使用SK对第二步的结果进行HMAC-SHA1签名计算。
- 4、对第三步的结果进行Base64编码，得到签名。

请求字符串(StringToSign)按照如下规则进行构造，各个参数的含义如表3-5所示。

```
StringToSign =  
    HTTP-Verb + "\n" +  
    Content-MD5 + "\n" +  
    Content-Type + "\n" +  
    Date + "\n" +  
    CanonicalizedHeaders + CanonicalizedResource
```

表 3-5 构造 StringToSign 所需参数说明

参数	描述
HTTP-Verb	指接口操作的方法，对REST接口而言，即为http请求操作的VERB，如： "PUT", "GET", "DELETE"等字符串。
Content-MD5	按照RFC 1864标准计算出消息体的MD5摘要字符串，即消息体128-bit MD5值经过base64编码后得到的字符串，可以为空。
Content-Type	内容类型，用于指定消息类型，例如： text/plain。 当请求中不带该头域时，该参数按照空字符串处理。

参数	描述
Date	生成请求的时间，该时间格式遵循RFC 1123；该时间与当前服务器的时间超过15分钟时服务端返回403。 当有自定义字段x-obs-date时，该参数按照空字符串处理。
Canonicalize dHeaders	HTTP请求头域中的SFS请求头字段，即以“x-obs-”作为前缀的头域，如“x-obs-date, x-obs-acl, x-obs-meta-*”。  1. 请求头字段中关键字的所有字符要转为小写（但内容值需要区分大小写，如“x-obs-storage-class:STANDARD”），需要添加多个字段时，要将所有字段按照关键字的字典序从小到大进行排序。  2. 在添加请求头字段时，如果有重名的字段，则需要合并。 如：x-obs-meta-name:name1和x-obs-meta-name:name2，则需要先将重名字段的值（这里是name1和name2）以逗号分隔，合并成x-obs-meta-name:name1,name2。  3. 头域中的请求头字段中的关键字不允许含有非ASCII码或不可识别字符；请求头字段中的值也不建议使用非ASCII码或不可识别字符，如果一定要使用非ASCII码或不可识别字符，需要客户端自行做编解码处理，可以采用URL编码或者Base64编码，服务端不会做解码处理。  4. 当请求头字段中含有无意义空格或table键时，需要摒弃。例如：x-obs-meta-name: name（name前带有一个无意义空格），需要转换为：x-obs-meta-name:name  5. 每一个请求头字段最后都需要另起新行。
Canonicalize dResource	表示HTTP请求所指定的SFS资源，构造方式如下： CanonicalizedResource = "/" + 文件系统名 + "?" + 子资源  1. 通用文件系统名，例如：/filesystem/。如果没有通用文件系统名，则为“/”。  2. 如果有子资源，则将子资源添加进来。 资源标识：sfsacl  3. 如果有多个子资源，在包含这些子资源时，需要首先将这些子资源按照其关键字的字典序从小到大排列，并使用“&”拼接。  <b>说明</b> 子资源通常是唯一的，不建议请求的URL包含多个相同关键字的子资源（例如，key=value1&key=value2），如果存在这种情况，SFS服务端签名时只会计算第一个子资源且也只有第一个子资源的值会对实际业务产生作用。

下面的提供了一些生成StringToSign的例子。

表 3-6 获取通用文件系统 ACL

请求消息头	StringToSign
GET /?sfsacl HTTP/1.1 Host: filesystem.sfs3.region.myhuaweicloud.com Date: Sat, 12 Oct 2015 08:12:38 GMT	GET \n \n \n Sat, 12 Oct 2015 08:12:38 GMT\n /filesystem/?sfsacl

## Java 中 Content-MD5 的计算方法示例

```
import java.security.MessageDigest;
import sun.misc.BASE64Encoder;
import java.io.UnsupportedEncodingException;
import java.security.NoSuchAlgorithmException;

public class Md5{
    public static void main(String[] args) {
        try {
            String exampleString = "blog";
            MessageDigest messageDigest = MessageDigest.getInstance("MD5");
            BASE64Encoder encoder = new BASE64Encoder();
            String contentMd5 = encoder.encode(messageDigest.digest(exampleString.getBytes("utf-8")));
            System.out.println("Content-MD5: " + contentMd5);
        } catch (NoSuchAlgorithmException | UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
}
```

根据请求字符串(StringToSign)和用户SK使用如下算法生成Signature，生成过程使用 HMAC算法(hash-based authentication code algorithm)。

Signature = Base64( HMAC-SHA1( YourSecretAccessKeyID, UTF-8-Encoding-Of( StringToSign ) ) )

例如在某区域创建通用文件系统名为newfilesystem2的通用文件系统，客户端请求格式为：

```
PUT / HTTP/1.1
Host: newfilesystem2.sfs3.region.myhuaweicloud.com
Content-Length: length
Date: Fri, 06 Jul 2018 03:45:51 GMT
x-obs-acl:private
x-obs-storage-class:STANDARD
Authorization: OBS UDSIAMSTUBTEST000254:ydH8ffpcbS6YpeOMcEZfn0wE90c=

<CreateBucketConfiguration xmlns="http://obs.myhwclouds.com/doc/2015-06-30/">
  <Location>region</Location>
</CreateBucketConfiguration>
```

## Java 中签名的计算方法

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Base64;
import java.util.Collections;
```

```
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.TreeMap;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

import org.omg.CosNaming.IStringHelper;

public class SignDemo {

    private static final String SIGN_SEP = "\n";

    private static final String SFS_PREFIX = "x-obs-";

    private static final String DEFAULT_ENCODING = "UTF-8";

    private static final List<String> SUB_RESOURCES = Collections.unmodifiableList(Arrays.asList(
        "CDNNotifyConfiguration", "acl", "append", "attname", "backtosource", "cors", "customdomain",
        "delete",
        "deletebucket", "directcoldaccess", "encryption", "inventory", "length", "lifecycle", "location",
        "logging",
        "metadata", "modify", "name", "notification", "orchestration", "partNumber", "policy", "position",
        "quota",
        "rename", "replication", "requestPayment", "response-cache-control", "response-content-
        disposition",
        "response-content-encoding", "response-content-language", "response-content-type", "response-
        expires",
        "restore", "select", "storageClass", "storagePolicy", "storageinfo", "tagging", "torrent", "truncate",
        "uploadId", "uploads", "versionId", "versioning", "versions", "website", "x-image-process",
        "x-image-save-bucket", "x-image-save-object", "x-obs-security-token"));

    private String ak;

    private String sk;

    public String urlEncode(String input) throws UnsupportedEncodingException
    {
        return URLEncoder.encode(input, DEFAULT_ENCODING)
            .replaceAll("%7E", "~") //for browser
            .replaceAll("%2F", "/")
            .replaceAll("%20", "+");
    }

    private String join(List<?> items, String delimiter)
    {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < items.size(); i++)
        {
            String item = items.get(i).toString();
            sb.append(item);
            if (i < items.size() - 1)
            {
                sb.append(delimiter);
            }
        }
        return sb.toString();
    }

    private boolean isValid(String input) {
        return input != null && !input.equals("");
    }

    public String hamcSha1(String input) throws NoSuchAlgorithmException, InvalidKeyException,
    UnsupportedEncodingException {
        SecretKeySpec signingKey = new SecretKeySpec(this.sk.getBytes(DEFAULT_ENCODING), "HmacSHA1");
```

```
Mac mac = Mac.getInstance("HmacSHA1");
mac.init(signingKey);
return Base64.getEncoder().encodeToString(mac.doFinal(input.getBytes(DEFAULT_ENCODING)));
}

private String stringToSign(String httpMethod, Map<String, String[]> headers, Map<String, String>
queries,
    String bucketName, String objectName) throws Exception{
    String contentMd5 = "";
    String contentType = "";
    String date = "";

    TreeMap<String, String> canonicalizedHeaders = new TreeMap<String, String>();

    String key;
    List<String> temp = new ArrayList<String>();
    for(Map.Entry<String, String[]> entry : headers.entrySet()) {
        key = entry.getKey();
        if(key == null || entry.getValue() == null || entry.getValue().length == 0) {
            continue;
        }

        key = key.trim().toLowerCase(Locale.ENGLISH);
        if(key.equals("content-md5")) {
            contentMd5 = entry.getValue()[0];
            continue;
        }

        if(key.equals("content-type")) {
            contentType = entry.getValue()[0];
            continue;
        }

        if(key.equals("date")) {
            date = entry.getValue()[0];
            continue;
        }

        if(key.startsWith(OBS_PREFIX)) {
            for(String value : entry.getValue()) {
                if(value != null) {
                    temp.add(value.trim());
                }
            }
            canonicalizedHeaders.put(key, this.join(temp, ","));
            temp.clear();
        }
    }

    if(canonicalizedHeaders.containsKey("x-obs-date")) {
        date = "";
    }

    // handle method/content-md5/content-type/date
    StringBuilder stringToSign = new StringBuilder();
    stringToSign.append(httpMethod).append(SIGN_SEP)
        .append(contentMd5).append(SIGN_SEP)
        .append(contentType).append(SIGN_SEP)
        .append(date).append(SIGN_SEP);

    // handle canonicalizedHeaders
    for(Map.Entry<String, String> entry : canonicalizedHeaders.entrySet()) {
        stringToSign.append(entry.getKey()).append(":").append(entry.getValue()).append(SIGN_SEP);
    }

    // handle CanonicalizedResource
    stringToSign.append("/");
}
```

```
        if(this.isValid(bucketName)) {
            stringToSign.append(bucketName).append("/");
            if(this.isValid(objectName)) {
                stringToSign.append(this.urlEncode(objectName));
            }
        }

        TreeMap<String, String> canonicalizedResource = new TreeMap<String, String>();
        for(Map.Entry<String, String> entry : queries.entrySet()) {
            key = entry.getKey();
            if(key == null) {
                continue;
            }

            if(SUB_RESOURCES.contains(key)) {
                canonicalizedResource.put(key, entry.getValue());
            }
        }

        if(canonicalizedResource.size() > 0) {
            stringToSign.append("?");
            for(Map.Entry<String, String> entry : canonicalizedResource.entrySet()) {
                stringToSign.append(entry.getKey());
                if(this.isValid(entry.getValue())) {
                    stringToSign.append("=").append(entry.getValue());
                }
            }
        }

//      System.out.println(String.format("StringToSign:%s%s", SIGN_SEP, stringToSign.toString()));

        return stringToSign.toString();
    }

    public String headerSignature(String httpMethod, Map<String, String[]> headers, Map<String, String>
queries,
        String bucketName, String objectName) throws Exception {

        //1. stringToSign
        String stringToSign = this.stringToSign(httpMethod, headers, queries, bucketName, objectName);

        //2. signature
        return String.format("OBS %s:%s", this.ak, this.hamcSha1(stringToSign));
    }

    public String querySignature(String httpMethod, Map<String, String[]> headers, Map<String, String>
queries,
        String bucketName, String objectName, long expires) throws Exception {
        if(headers.containsKey("x-obs-date")) {
            headers.put("x-obs-date", new String[] {String.valueOf(expires)});
        }else {
            headers.put("date", new String[] {String.valueOf(expires)});
        }
        //1. stringToSign
        String stringToSign = this.stringToSign(httpMethod, headers, queries, bucketName, objectName);

        //2. signature
        return this.urlEncode(this.hamcSha1(stringToSign));
    }

    public static void main(String[] args) throws Exception {

        SignDemo demo = new SignDemo();
        demo.ak = "<your-access-key-id>";
        demo.sk = "<your-secret-key-id>";

        String bucketName = "bucket-test";
        String objectName = "hello.jpg";
    }
}
```

```
Map<String, String[]> headers = new HashMap<String, String[]>();
headers.put("date", new String[] {"Sat, 12 Oct 2015 08:12:38 GMT"});
headers.put("x-obs-acl", new String[] {"private"});
Map<String, String> queries = new HashMap<String, String>();
queries.put("acl", null);

System.out.println(demo.headerSignature("PUT", headers, queries, bucketName, objectName));
}
}
```

签名计算的样例结果为（按照执行时间的不同变化）：  
ydH8ffpcbS6YpeOMcEZfn0wE90c=

## Python 中签名的计算方法

```
import sys
import os
import hashlib
import hmac
import binascii
from datetime import datetime

IS_PYTHON2 = sys.version_info.major == 2 or sys.version < '3'
"""认证用的SecretAccessKeyID硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；本示例以SecretAccessKeyID保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量SECRET_ACCESS_KEY_ID。"""
yourSecretAccessKeyID = os.environ.get('SECRET_ACCESS_KEY_ID')
httpMethod = "PUT"
contentType = "application/xml"
httpMethod = "PUT"
contentType = "application/xml"
# "date" is the time when the request was actually generated
date = datetime.utcnow().strftime('%a, %d %b %Y %H:%M:%S GMT')
canonicalizedHeaders = "x-obs-acl:private\n"
CanonicalizedResource = "/newfilesystem2"
canonical_string = httpMethod + "\n" + "\n" + contentType + "\n" + date + "\n" + canonicalizedHeaders + CanonicalizedResource
if IS_PYTHON2:
    hashed = hmac.new(yourSecretAccessKeyID, canonical_string, hashlib.sha1)
    encode_canonical = binascii.b2a_base64(hashed.digest())[:-1]
else:
    hashed = hmac.new(yourSecretAccessKeyID.encode('UTF-8'), canonical_string.encode('UTF-8'), hashlib.sha1)
    encode_canonical = binascii.b2a_base64(hashed.digest())[:-1].decode('UTF-8')
print encode_canonical
```

签名计算的样例结果为（按照执行时间的不同变化）：  
ydH8ffpcbS6YpeOMcEZfn0wE90c=

## C 语言中签名的计算方法

请单击[此处](#)，下载C语言签名计算代码样例，其中：

1. 计算签名的接口包含在sign.h头文件中。
2. 计算签名的示例代码在main.c文件中。

## 签名不匹配报错处理

若调用SFS API报如下错误：

状态码：403 Forbidden

错误码：SignatureDoesNotMatch

错误信息：The request signature we calculated does not match the signature you provided. Check your key and signing method.

请联系技术支持。

### 3.2.3 签名计算工具

SFS提供可视化签名计算工具，帮助您轻松完成签名计算。

表 3-7 签名计算工具

签名计算方式	签名计算工具获取地址
Header中携带签名	<a href="#">可视化签名计算工具</a>

若调用SFS API报如下错误：

状态码：403 Forbidden

错误码：SignatureDoesNotMatch

错误信息：The request signature we calculated does not match the signature you provided. Check your key and signing method.

请联系技术支持。

## 3.3 返回结果

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

### 状态码

状态码是一组从2xx（成功）到4xx或5xx（错误）的数字代码，状态码表示了请求响应的状态，完整的状态码列表请参见[状态码](#)。

### 响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

详细的公共响应消息头字段请参见[表3-8](#)。

表 3-8 公共响应消息头

消息头名称	描述
Content-Length	响应消息体的字节长度。 类型：字符串。 默认值：无。

消息头名称	描述
Connection	指明与服务器的连接是长连接还是短连接。 类型：字符串。 有效值：keep-alive   close。 默认值：无。
Date	系统响应的时间。 类型：字符串。 默认值：无。
x-obs-id-2	帮助定位问题的特殊符号。 类型：字符串。 默认值：无。
x-obs-request-id	由SFS创建来唯一确定本次请求的值，可以通过该值来定位问题。 类型：字符串。 默认值：无。

响应消息体（可选）

响应消息体通常以结构化格式（如JSON或XML）返回，与响应消息头中Content-type对应，传递除响应消息头之外的内容。

# 4 快速入门

## 4.1 创建通用文件系统

### 操作场景

通用文件系统是SFS中存储文件的容器。您需要先创建一个通用文件系统，然后才能在SFS中存储数据。

下面介绍如何调用[创建通用文件系统 - CreateFileSystem](#) API在指定的区域创建一个通用文件系统，API的调用方法请参见[如何调用API](#)。

### 前提条件

- 已获取AK和SK，获取方法参见[获取访问密钥（AK/SK）](#)。
- 您需要规划通用文件系统所在的区域信息，并根据区域确定调用API的Endpoint，详细信息请参见[终端节点](#)。

区域一旦确定，创建完成后无法修改。

### 在 a1 区域创建一个名为 filesystem001 的通用文件系统

示例中使用通用的Apache Http Client。

```
package com.sfsclient;

import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.List;
import java.util.Map;

public class TestMain {
    // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
    // 您可以登录访问管理控制台获取访问密钥AK/SK
    public static String accessKey = System.getenv("HUAWEICLOUD_SDK_AK");
    public static String securityKey = System.getenv("HUAWEICLOUD_SDK_SK");
    public static String region = "cn-east-3"; // 取值为规划通用文件系统所在的区域

    public static String endpoint = "sfs3.a1.region.com"; // 通用文件系统服务地址

    public static String createSfsBody =
```

```
"<CreateBucketConfiguration>\n" +
"<Location>" + region + "</Location>\n" +
"</CreateBucketConfiguration>";

public static void main(String[] str) {

    createFileSystem();

}
private static void createFileSystem() {
    // 通用文件系统名称
    String fileName = "example-sfs-001";

    String httpMethod = "PUT";
    String date = DateUtils.formatDate(System.currentTimeMillis());
    String contentType = "application/xml";

    /** 根据请求计算签名**/
    String contentMD5 = "";
    String canonicalizedHeaders = "x-obs-bucket-type:SFS";
    String canonicalizedResource = "/" + fileName;

    // Content-MD5 、 Content-Type 没有直接换行，data格式为RFC 1123，和请求中的时间一致
    String stringToSign = httpMethod + "\n" +
        contentMD5 + "\n" +
        contentType + "\n" +
        date + "\n" +
        canonicalizedHeaders + "\n" + canonicalizedResource;

    System.out.printf("StringToSign:\n[%s]\n\n", stringToSign);

    HttpURLConnection conn = null;

    try {
        String signature = Signature.signWithHmacSha1(securityKey, stringToSign);
        String authorization = "OBS " + accessKey + ":" + signature;
        System.out.printf("authorization:%s\n\n", authorization);

        URL url = new URL("http://" + endpoint + "/" + fileName);
        conn = (HttpURLConnection) url.openConnection();

        // 增加签名头域
        conn.setRequestMethod(httpMethod);
        conn.setRequestProperty("Date", date);
        conn.setRequestProperty("Content-Type", contentType);
        conn.setRequestProperty("x-obs-bucket-type", "SFS");
        conn.setRequestProperty("Authorization", authorization);
        conn.setDoOutput(true);

        // 增加body体
        OutputStream out = conn.getOutputStream();
        out.write(createSfsBody.getBytes());
        out.flush();
        out.close();

        String status = conn.getHeaderField(null);
        System.out.println(status);

        // 输出收到的响应消息
        Map<String, List<String>> headers = conn.getHeaderFields();
        for (Map.Entry<String, List<String>> entry : headers.entrySet()) {
            String key = entry.getKey();
            List<String> values = entry.getValue();
            if (key != null) {
                for (String value : values) {
                    System.out.println(key + ": " + value);
                }
            }
        }
    }
}
```

```
// 请求出错处理
int statusCode = conn.getResponseCode();
if (statusCode != HttpURLConnection.HTTP_OK && statusCode !=
HttpURLConnection.HTTP_NO_CONTENT) {
    InputStream errorStream = conn.getErrorStream();
    BufferedReader reader = new BufferedReader(new InputStreamReader(errorStream));
    StringBuilder responseBody = new StringBuilder();
    String line;
    while ((line = reader.readLine()) != null) {
        responseBody.append(line);
    }
    reader.close();

    System.out.println("Error: " + responseBody);
}
} catch (IOException e) {
    e.printStackTrace();
} finally {
    if (conn != null){
        conn.disconnect();
    }
}
}
```

其中Date头域DateUtils的格式为:

```
package com.sfsclient;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Locale;
import java.util.TimeZone;

public class DateUtils {

    public static String formatDate(long time)
    {
        DateFormat serverDateFormat = new SimpleDateFormat("EEE, dd MMM yyyy HH:mm:ss z",
Locale.ENGLISH);
        serverDateFormat.setTimeZone(TimeZone.getTimeZone("GMT"));
        return serverDateFormat.format(time);
    }
}
```

签名字符串Signature的计算方法为:

```
package com.sfsclient;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.io.UnsupportedEncodingException;
import java.security.NoSuchAlgorithmException;
import java.security.InvalidKeyException;
import java.util.Base64;

public class Signature {

    public static String signWithHmacSha1(String sk, String canonicalString) throws
UnsupportedEncodingException {

        try {
            SecretKeySpec signingKey = new SecretKeySpec(sk.getBytes("UTF-8"), "HmacSHA1");
            Mac mac = Mac.getInstance("HmacSHA1");
            mac.init(signingKey);
            return Base64.getEncoder().encodeToString(mac.doFinal(canonicalString.getBytes("UTF-8")));
        } catch (NoSuchAlgorithmException | InvalidKeyException | UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
}
```

```
        return null;
    }
}
```

## 4.2 获取通用文件系统列表

### 操作场景

如果用户想要查看自己创建的所有通用文件系统信息，可以使用获取通用文件系统列表接口查看。

下面介绍如何调用[获取通用文件系统列表 - ListFileSystems](#)API，API的调用方法请参见[快速入门](#)。

### 前提条件

- 已获取AK和SK，获取方法参见[获取访问密钥（AK/SK）](#)。
- 您需要规划通用文件系统所在的区域信息，并根据区域确定调用API的Endpoint，详细信息请参见[终端节点](#)。

### 获取 a1 区域的通用文件系统列表

示例中使用通用的Apache Http Client。

```
package com.sfsclient;

import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.List;
import java.util.Map;

public class TestMain {
    // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
    // 您可以登录访问管理控制台获取访问密钥AK/SK
    public static String accessKey = "ACCESS_KEY_ID";
    public static String securityKey = "SECRET_ACCESS_KEY_ID";
    public static String endpoint = "sfs3.a1.region.com"; // 通用文件系统服务地址

    public static void main(String[] str) {
        createFileSystem();
    }

    private static void listFileSystem() {
        String httpMethod = "GET";
        String date = DateUtils.formatDate(System.currentTimeMillis());

        /** 根据请求计算签名**/
        String contentMD5 = "";
        String contentType = "";
        String canonicalizedHeaders = "x-obs-bucket-type:SFS";
        String canonicalizedResource = "/";

        // Content-MD5、Content-Type 没有直接换行，data格式为RFC 1123，和请求中的时间一致
        String stringToSign = httpMethod + "\n" +
            contentMD5 + "\n" +
            contentType + "\n" +
            date + "\n" +
            canonicalizedHeaders + "\n" + canonicalizedResource;

        System.out.printf("StringToSign:\n[%s]\n\n", stringToSign);
    }
}
```

```
URLConnection conn = null;

try {
    String signature = Signature.signWithHmacSha1(securityKey, stringToSign);
    String authorization= "OBS " + accessKey + ":" + signature;
    System.out.printf("authorization:%s\n\n", authorization);

    // 创建http请求
    URL url = new URL("http://" + endpoint);
    conn = (URLConnection) url.openConnection();

    // 增加签名头域
    conn.setRequestMethod(httpMethod);
    conn.setRequestProperty("Date", date);
    conn.setRequestProperty("Content-Type", contentType);
    conn.setRequestProperty("x-obs-bucket-type", "SFS");
    conn.setRequestProperty("Authorization", authorization);
    conn.setDoOutput(true);

    String status = conn.getHeaderField(null);
    System.out.println(status);

    // 输出收到的响应消息
    Map<String, List<String>> headers = conn.getHeaderFields();
    for (Map.Entry<String, List<String>> entry : headers.entrySet()) {
        String key = entry.getKey();
        List<String> values = entry.getValue();
        if (key != null) {
            for (String value : values) {
                System.out.println(key + ": " + value);
            }
        }
    }
    // 请求返回内容处理
    int statusCode = conn.getResponseCode();
    if (statusCode == HttpURLConnection.HTTP_OK) {
        InputStream responseStream = conn.getInputStream();
        BufferedReader reader = new BufferedReader(new InputStreamReader(responseStream));

        StringBuilder responseBody = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            responseBody.append(line);
        }
        reader.close();

        System.out.println("responseBody: " + responseBody);
    } else {
        System.out.println("Error: " + statusCode);
    }
} catch (IOException e) {
    e.printStackTrace();
} finally {
    if (conn != null){
        conn.disconnect();
    }
}
}
```

其中Date头域DateUtils的格式为:

```
package com.sfsclient;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Locale;
import java.util.TimeZone;
```

```
public class DateUtils {

    public static String formatDate(long time)
    {
        DateFormat serverDateFormat = new SimpleDateFormat("EEE, dd MMM yyyy HH:mm:ss z",
Locale.ENGLISH);
        serverDateFormat.setTimeZone(TimeZone.getTimeZone("GMT"));
        return serverDateFormat.format(time);
    }
}
```

### 签名字符串Signature的计算方法为：

```
package com.sfsclient;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.io.UnsupportedEncodingException;
import java.security.NoSuchAlgorithmException;
import java.security.InvalidKeyException;
import java.util.Base64;

public class Signature {
    public static String signWithHmacSha1(String sk, String canonicalString) throws
UnsupportedEncodingException {

        try {
            SecretKeySpec signingKey = new SecretKeySpec(sk.getBytes("UTF-8"), "HmacSHA1");
            Mac mac = Mac.getInstance("HmacSHA1");
            mac.init(signingKey);
            return Base64.getEncoder().encodeToString(mac.doFinal(canonicalString.getBytes("UTF-8")));
        } catch (NoSuchAlgorithmException | InvalidKeyException | UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

# 5 API 说明

## 5.1 通用文件系统管理

### 5.1.1 创建通用文件系统 - CreateFileSystem

#### 功能介绍

创建通用文件系统是指按照用户指定的通用文件系统名称创建一个新通用文件系统的操作。

 说明

- 新创建文件系统的名称在SFS中必须是唯一的。如果是同一个用户重复创建同一区域的同名文件系统时将返回成功。除此以外的其他场景重复创建同名文件系统返回文件系统已存在。
- 默认情况下，一个用户可以拥有的文件系统的数量不能超过100个。
- 用户删除文件系统后，需要等待30分钟才能创建同名文件系统。

#### 授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

#### URI

PUT /

#### 请求参数

表 5-1 请求 Header 参数

参数	是否必选	参数类型	描述
Authorization	是	String	签名信息
Date	是	String	请求时间

参数	是否必选	参数类型	描述
x-obs-az-redundancy	否	String	单AZ还是多AZ
x-obs-bucket-type	是	String	通过此消息头明确创建通用文件系统。取值：SFS：创建通用文件系统。
Host	是	String	表明主机地址。
x-obs-epid	否	String	企业项目ID

表 5-2 请求 Body 参数

参数	是否必选	参数类型	描述
Location	否	String	区域

响应参数

该请求的响应消息使用公共消息头，具体请参考 表3-8。

响应消息体（可选）

响应消息体通常以结构化格式（如JSON或XML）返回，与响应消息头中Content-type对应，传递除响应消息头之外的内容。

请求示例

- 在“example”区域创建通用文件系统，主机地址是Host: example-sfs-01.sfs3.example.region.com:443，企业项目ID为“0”。

```
PUT / HTTP/1.1
Host: example-sfs-01.sfs3.example.region.com:443
Date: Wed, 07 Jun 2023 02:38:09 GMT
x-obs-bucket-type: SFS
Authorization: OBS FNEX1B77SXDIB3TFMSZZ:0Xsnu4hJVOI7VWH0wIQczVN+rbg=
Content-Length: 85
x-obs-epid: 0
<CreateBucketConfiguration>
  <Location>example</Location>
</CreateBucketConfiguration>
```

响应示例

```
HTTP/1.1 200 OK
Server: OBS
X-Obs-Request-Id: 0000018893B8058EC0470388BE6EDE88
Location: /example-sfs-01
X-Obs-Id-2: 32AAQAEEAABSAAGAAEAABAAAQAAEAABCTRa4voOUvr50ncznQT/hligMxL4so2z
Date: Wed, 07 Jun 2023 02:38:11 GMT
Content-Length: 0
```

状态码

状态码	描述
200	创建通用文件系统成功

错误码

请参见[错误码](#)。

5.1.2 删除通用文件系统 - DeleteFileSystem

功能介绍

删除通用文件系统。

授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

URI

DELETE /

请求参数

表 5-3 请求 Header 参数

参数	是否必选	参数类型	描述
Authorization	是	String	签名头域
Date	是	String	请求时间
Host	是	String	表明主机地址。

响应参数

该请求的响应消息使用公共消息头，具体请参考 [表3-8](#)。

请求示例

```
DELETE / HTTP/1.1
User-Agent: curl/7.29.0
Host: examplefilesystem.sfs3.example.region.com
Accept: */*
Date: WED, 01 Jul 2015 02:31:25 GMT
Authorization: OBS H4lPJX0TQTHTHEBQQCEC:jZiAT8Vx4azWEvPRMWi0X5BpJMA=

HTTP/1.1 204 No Content
Server: OBS
```

```
X-Obs-Request-Id: 0000018893B8081DC047305E783867DD
X-Obs-Id-2: 32AAQAEEAABSkAgAAEAABAAQAEEAABCT5UWgsaro3EEEnOsNEzf8w8dnydR+Eak
Date: WED, 01 Jul 2015 02:31:25 GMT
```

状态码

状态码	描述
204	删除通用文件系统成功

错误码

请参见[错误码](#)。

5.1.3 获取通用文件系统列表 - ListFileSystems

功能介绍

获取通用文件系统列表。

授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

URI

GET /

请求参数

表 5-4 请求 Header 参数

参数	是否必选	参数类型	描述
x-obs-bucket-type	是	String	通过此消息头明确获取的列表内容。取值：SFS：获取所有通用文件系统列表。
Authorization	是	String	签名信息
Date	是	String	请求时间
Host	是	String	表明主机地址。

响应参数

状态码： 200

表 5-5 响应 Body 参数

参数	参数类型	描述
Owner	<b>Owner</b> object	通用文件系统拥有者信息，包含租户ID
Buckets	<b>Buckets</b> object	用户所拥有的通用文件系统列表

表 5-6 Owner

参数	参数类型	描述
ID	String	用户的DomainID（账号ID）

表 5-7 Buckets

参数	参数类型	描述
Bucket	<b>Bucket</b> object	具体的通用文件系统信息

表 5-8 Bucket

参数	参数类型	描述
Name	String	通用文件系统名称
CreationDate	String	通用文件系统的创建时间
Location	String	通用文件系统的位置信息

请求示例

```
GET / HTTP/1.1
Host: sfs3.example.region.com
Date: date
x-obs-bucket-type: SFS
Authorization: authorization
```

响应示例

```
HTTP/1.1 200 OK
Server: OBS
X-Obs-Request-Id: 0000018893B8126DC048B06DD3816BD4
X-Obs-Id-2: 32AAQAEEAABAAAQAEEAABAAAQAEEAABCTMZh3Thi7lcDxuGWu9QtP9PjYx7lb
Date: Wed, 07 Jun 2023 02:38:14 GMT
Content-Type: application/xml
Content-Length: 377

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ListAllMyBucketsResult xmlns="http://obs.myhwclouds.com/doc/2015-06-30/">
  <Owner>
    <ID>783fc6652cf246c096ea836694f71855</ID>
```

```
</Owner>
<Buckets>
  <Bucket>
    <Name>examplebucket01</Name>
    <CreationDate>2018-06-21T09:15:01.032Z</CreationDate>
    <Location>example-region-1</Location>
    <BucketType>SFS</BucketType>
  </Bucket>
  <Bucket>
    <Name>examplebucket02</Name>
    <CreationDate>2018-06-22T03:56:33.700Z</CreationDate>
    <Location>example-region-2</Location>
    <BucketType>SFS</BucketType>
  </Bucket>
</Buckets>
</ListAllMyBucketsResult>
```

## 状态码

状态码	描述
200	成功响应消息样式，获取通用文件系统列表成功

## 错误码

请参见[错误码](#)。

## 5.2 访问规则管理

### 5.2.1 设置通用文件系统 ACL - PutACL

#### 功能介绍

设置通用文件系统ACL。

##### 说明

通用文件系统ACL设置生效会有30s的滞后时间。

#### 授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

#### URI

PUT /

表 5-9 Query 参数

参数	是否必选	参数类型	描述
sfsacl	是	String	/

请求参数

表 5-10 请求 Header 参数

参数	是否必选	参数类型	描述
Date	是	String	请求消息时间
Authorization	是	String	请求签名信息
Host	是	String	表明主机地址。

表 5-11 请求 Body 参数

参数	是否必选	参数类型	描述
Statement	否	Array of <a href="#">Statement</a> objects	Unique identification

表 5-12 Statement

参数	是否必选	参数类型	描述
Sid	否	String	statement Id，描述statement的字符串
Action	是	String	本条statement作用的操作，取值范围：FullControl（读写）、Read（只读）。 枚举值： <ul style="list-style-type: none"><li>FullControl</li><li>Read</li></ul>
Effect	是	String	效果，指定本条statement的权限是允许，必须为Allow。 枚举值： <ul style="list-style-type: none"><li>Allow</li></ul>
Condition	是	<a href="#">Condition</a> object	本条statement生效的条件。

表 5-13 Condition

参数	是否必选	参数类型	描述
SourceVpc	是	String	指定VPC的VPCid。
VpcSourceIp	否	Array of strings	指定的IP或IP范围（暂不支持该参数）。

响应参数

该请求的响应消息使用公共消息头，具体请参考 [表3-8](#)。

请求示例

设置通用文件系统ACL，授予读写权限，VPC ID为“241dbf6b-dc5d-41b2-9108-ca5e56b48386”，IP范围为["127.0.0.1/24", "192.168.1.85/24"]。

```
PUT /?sfsacl HTTP/1.1
Host: examplefilesystem.sfs3.example.region.com
Date: WED, 01 Jul 2015 02:32:25 GMT
Authorization: OBS H4lPJX0TQTHTHEBQQCEC:jZiAT8Vx4azWEvPRMWi0X5BpJMA=

{
  "Statement": [{
    "Sid": "Stmt1375240018061",
    "Action": "FullControl",
    "Effect": "Allow",
    "Condition": {
      "SourceVpc": "241dbf6b-dc5d-41b2-9108-ca5e56b48386",
      "VpcSourceIp": ["127.0.0.1/24", "192.168.1.85/24"]
    }
  }]
}
```

响应示例

```
HTTP/1.1 204 OK
Server: OBS
X-Obs-Request-Id: 0000018893B8073AC04721AA7EE3408B
X-Obs-Id-2: 32AAQAEEAABSAAGAAEAABAAQAEEAABCS5QDe0QLbFNz6FXoKuXHzD2wS0eJQaj
Date: Wed, 07 Jun 2023 02:38:11 GMT
```

状态码

状态码	描述
204	设置通用文件系统ACL成功

错误码

请参见[错误码](#)。

## 5.2.2 获取通用文件系统 ACL - GetACL

### 功能介绍

获取通用文件系统ACL。

### 授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

### URI

GET /

表 5-14 Query 参数

参数	是否必选	参数类型	描述
sfsacl	是	String	/

### 请求参数

表 5-15 请求 Header 参数

参数	是否必选	参数类型	描述
Date	是	String	请求消息时间
Authorization	是	String	请求签名信息
Host	是	String	表明主机地址。

### 响应参数

状态码： 200

表 5-16 响应 Body 参数

参数	参数类型	描述
Statement	Array of <a href="#">Statement</a> objects	Unique identification

表 5-17 Statement

参数	参数类型	描述
Sid	String	statement Id，描述statement的字符串
Action	String	本条statement作用的操作，取值范围：FullControl（读写）、Read（只读）。 枚举值： <ul style="list-style-type: none"><li>FullControl</li><li>Read</li></ul>
Effect	String	效果，指定本条statement的权限是允许，必须为Allow。 枚举值： <ul style="list-style-type: none"><li>Allow</li></ul>
Condition	Condition object	本条statement生效的条件。

表 5-18 Condition

参数	参数类型	描述
SourceVpc	String	指定VPC的VPCid。
VpcSourceIp	Array of strings	指定的IP或IP范围（暂不支持该参数）。

请求示例

```
GET /?sfsacl HTTP/1.1
Host: example-sfs-01.sfs3.example.region.com:443
Date: Wed, 07 Jun 2023 03:31:46 GMT
Authorization: OBS FNEX1B77SXDIB3TFMSZZ:eUqPIHnPDWGDTlgyLmsALA86wys=
```

响应示例

```
HTTP/1.1 200 OK
Server: OBS
Content-Type: application/json
Content-Length: 131
Date: Wed, 07 Jun 2023 03:31:59 GMT
X-Obs-Request-Id: 0000018893E94B65C046B527778F8F14
X-Obs-Id-2: 32AAQAEEAABAAQAEEAABAAQAEEAABCSc2lEdShcA04319WknB1DD5BdBKuGr1
{
  "Statement": [
    {
      "Condition": {
        "SourceVpc": "f85adabc-a387-4d1d-94cf-65ef9034f752"
      },
      "Action": "FullControl",
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
```

```
]
}
```

状态码

状态码	描述
200	获取通用文件系统ACL成功

错误码

请参见[错误码](#)。

5.2.3 删除通用文件系统 ACL - DeleteACL

功能介绍

删除通用文件系统ACL。

授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

URI

DELETE /

表 5-19 Query 参数

参数	是否必选	参数类型	描述
sfsacl	是	String	/

请求参数

表 5-20 请求 Header 参数

参数	是否必选	参数类型	描述
Date	是	String	请求消息时间
Authorization	是	String	请求签名信息
Host	是	String	表明主机地址。

响应参数

该请求的响应消息使用公共消息头，具体请参考 [表3-8](#)。

请求示例

```
DELETE /?sfsacl HTTP/1.1
Host: examplefilesystem.sfs3.example.region.com
Date: WED, 01 Jul 2015 02:36:06 GMT
Authorization: OBS H4lPJX0TQTHTHEBQQCEC;jZiAT8Vx4azWEvPRMWi0X5BpJMA=
```

响应示例

```
HTTP/1.1 204 No Content
Server: OBS
X-Obs-Id-2: 32AAAQAAEAABSAAGAAEAABAAQAAEAABCSj4dxiqb1Lw50CTjVQeV3ebh3QQ6PAj
X-Obs-Request-Id: 0000018893B807D5C0472A6161D87032
Date: WED, 01 Jul 2015 02:36:06 GMT
```

状态码

状态码	描述
204	删除通用文件系统ACL成功

错误码

请参见[错误码](#)。

5.3 标签管理

5.3.1 批量添加资源标签 - TagResources

功能介绍

为指定资源（资源指通用文件系统）批量添加标签。一个资源上最多有20个标签。

授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

URI

- POST /v3/sfs/tms/{project\_id}/file-systems/{resource\_id}/tags/create
- 参数说明

参数	是否必选	参数类型	描述
project_id	是	String	操作用户的项目ID，获取方法请参见 <a href="#">获取项目ID</a> 。
resource_id	是	String	资源ID，此处为通用文件系统名。



参数	是否必选	参数类型	描述
value	是	String	标签的值。最大长度为255个字符，首尾不支持空格字符，值可以为空字符串，可用UTF-8格式表示的任意语种的字母、数字和空格，以及以下字符：“ ”、“.”、“=”、“+”、“-”、“@”。

响应参数

无

请求示例

projectId为 “c80a2157ba1d46c0825265947342077c” ， 通用文件系统名为 “bucketName” ， 批量添加资源标签示例：

POST https://{endpoint}/v3/sfs/tms/c80a2157ba1d46c0825265947342077c/file-systems/bucketName/tags/create

请求体示例：

```
{
  "tags":[
    {
      "key":"key1",
      "value":"value1"
    },
    {
      "key":"key2",
      "value":"value2"
    }
  ]
}
```

响应示例

无

状态码

- 正常

状态码	描述
204	批量添加资源标签成功。

- 异常

状态码	描述
400	标签等参数非法。

状态码	描述
401	认证失败。
403	鉴权失败。
404	资源未找到。
500	系统异常。

### 5.3.2 批量删除资源标签 - UnTagResources

#### 功能介绍

为指定资源批量删除标签。不支持删除系统标签。删除时，如果删除的标签不存在，默认处理成功。

#### 授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

#### URI

- POST /v3/sfs/tms/{project\_id}/file-systems/{resource\_id}/tags/delete
- 参数说明

参数	是否必选	参数类型	描述
project_id	是	String	操作用户的项目ID，获取方法请参见 <a href="#">获取项目ID</a> 。
resource_id	是	String	资源ID，此处为通用文件系统名。

#### 请求参数

表 5-24 请求 Header 参数

参数	是否必选	参数类型	描述
Content-type	是	String	发送的实体的MIME类型。 示例：application/json。
X-Auth-Token	否	String	用户Token。

表 5-25 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	List<resource_tag>	标签列表。详情说明参见表 5-23。

响应参数

无

请求示例

projectId为“c80a2157ba1d46c0825265947342077c”，通用文件系统名为“bucketName”，批量删除资源标签示例：

POST https://{endpoint}/v3/sfs/tms/c80a2157ba1d46c0825265947342077c/file-systems/bucketName/tags/delete

请求体示例：

```
{
  "tags": [
    {
      "key": "key1"
    },
    {
      "key": "key2",
      "value": "value2"
    }
  ]
}
```

响应示例

无

状态码

- 正常

状态码	描述
204	批量删除资源标签成功。

- 异常

状态码	描述
400	标签等参数非法。
401	认证失败。
403	鉴权失败。

状态码	描述
404	资源未找到。
500	系统异常。

### 5.3.3 查询资源标签 - ListTagsForResource

#### 功能介绍

查询指定资源的标签。

#### 授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

#### URI

- GET /v3/sfs/tms/{project\_id}/file-systems/{resource\_id}/tags
- 参数说明

参数	是否必选	参数类型	描述
project_id	是	String	操作用户的项目ID，获取方法请参见 <a href="#">获取项目ID</a> 。
resource_id	是	String	资源ID，此处为通用文件系统名。

#### 请求参数

表 5-26 请求 Header 参数

参数	是否必选	参数类型	描述
Content-type	是	String	发送的实体的MIME类型。 示例：application/json。
X-Auth-Token	否	String	用户Token。

响应参数

表 5-27 响应 Body 参数

参数	是否必选	参数类型	描述
tags	否	List<resource_tag>	标签列表。详情说明参见表 5-23。
sys_tags	否	List<resource_tag>	系统标签列表。仅op_service权限才可以获取此字段。 目前只包含一个resource_tag结构体： key： _sys_enterprise_project_id value：企业项目id。0表示默认企业项目。 详情说明参见表5-23。

请求示例

projectId为 “c80a2157ba1d46c0825265947342077c” ， 通用文件系统名为 “bucketName” ， 查询资源标签示例：

GET https://{endpoint}/v3/sfs/tms/c80a2157ba1d46c0825265947342077c/file-systems/bucketName/tags

响应示例

```
{
  "tags": [
    {
      "key": "key1",
      "value": "value1"
    },
    {
      "key": "key2",
      "value": "value2"
    }
  ]
}
```

状态码

- 正常

状态码	描述
200	查询资源标签成功。

- 异常

状态码	描述
400	标签等参数非法
401	认证失败
403	鉴权失败
404	资源未找到
500	系统异常

### 5.3.4 查询资源实例列表 - ListResourcesByTag

#### 功能介绍

使用标签过滤资源实例。优先按照资源创建时间倒序排序。

#### 授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

#### URI

- POST /v3/sfs/tms/{project\_id}/file-systems/resource-instances/filter
- 参数说明

参数	是否必选	参数类型	描述
project_id	是	String	操作用户的项目ID，获取方法请参见 <a href="#">获取项目ID</a> 。
limit	否	Int	查询记录数。默认为1000，limit最大为1000，最小值为1，不能为负数。
offset	否	Int	索引位置。从第一条数据偏移offset条数据后开始查询。默认为0（偏移0条数据，表示从第一条数据开始查询），必须为数字，不能为负数。

#### 请求参数

表 5-28 请求 Header 参数

参数	是否必选	参数类型	描述
Content-type	是	String	发送的实体的MIME类型。 示例：application/json。

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。

表 5-29 请求 Body 参数

参数	是否必选	参数类型	描述
without_any_tag	否	boolean	不包含任意一个标签。该字段为true时查询所有不带标签的资源，此时忽略“tags”字段。该字段为false或者未提供该参数时，该条件不生效，即返回所有资源或按"tags"，"matches"条件过滤的资源。
tags	否	List<tag>	包含标签。最多包含20个tag。tag之间Key不能重复，每个key下面的value最多20个，每个key对应的value可以为空数组但结构体不能缺失，且同一个key中values不能重复。 结果返回包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无tag过滤条件时返回全量数据。 详细说明参见 <a href="#">表5-30</a> 。
sys_tags	否	List<tag>	包含系统标签。仅op_service权限可以使用此字段。 sys_tags和过滤条件without_any_tag、tags不能同时使用。 目前此字段只包含一个tag结构体： key： _sys_enterprise_project_id value：企业项目id列表 目前key下面只包含一个value。0表示默认企业项目。 详细说明参见 <a href="#">表5-30</a> 。
matches	否	List<match>	搜索字段。match中key为固定字典值，当前key固定为resource_name，表示根据key对应的value值实现前缀搜索，后续再扩展。 详细说明参见 <a href="#">表5-31</a> 。

表 5-30 tag

参数	是否必选	参数类型	描述
key	是	String	标签的键。最大长度为128个字符，不能为空，首尾不支持空格字符，以“_sys_”开头的键属于系统标签，不支持用户输入，可用UTF-8格式表示的任意语种的字母、数字和空格，以及“-”、“.”、“=”、“+”、“_”、“@”。
values	是	List<String>	值列表。value可为空数组但不可缺省。如果values为空列表，则表示查询任意value。value之间为或的关系。

表 5-31 match

参数	是否必选	参数类型	描述
key	是	String	键。当前限定为resource_name，后续扩展。
value	是	String	值。每个值最大长度255个Unicode字符。不校验字符集范。

响应参数

表 5-32 响应 body 参数

参数	是否必选	参数类型	描述
resources	是	List< <a href="#">resource</a> >	资源实例列表。详情说明参见 <a href="#">表 5-33</a> 。
total_count	是	Integer	总记录数。

表 5-33 resource

参数	是否必选	参数类型	描述
resource_id	是	String	资源ID。
resource_detail	是	Object	资源详情。默认为空。资源对象，用于扩展。

参数	是否必选	参数类型	描述
tags	是	List<resource_tag>	标签列表。没有标签默认为空数组。 详情说明参见表5-23。
sys_tags	否	List<resource_tag>	仅op_service权限才可以获取此字段。 目前只包含一个resource_tag 结构体： key： _sys_enterprise_project_id value：企业项目id，0表示默认企业项目。 详情说明参见表5-23。
resource_name	是	String	资源名称，资源没有名称时默认为空字符串。

请求示例

projectId为 “c80a2157ba1d46c0825265947342077c” ， offset为0， limit为10， 查询资源实例示例：

POST https://{endpoint}/v3/sfs/tms/c80a2157ba1d46c0825265947342077c/file-systems/resource-instances/filter?limit=10&offset=0

请求体示例：

```
{
  "tags": [
    {
      "key": "key1",
      "values": [
        "value1",
        "value2"
      ]
    },
    {
      "key": "key2",
      "values": [
        "value1",
        "value2"
      ]
    }
  ],
  "matches": [
    {
      "key": "resource_name",
      "value": "resource1"
    }
  ],
  "without_any_tag": "false"
}
```

响应示例

```
{
  "resources": [
```

```
{
  "resource_detail":"","
  "resource_id":"resource1",
  "resource_name":"resource1",
  "tags":[
    {
      "key":"key1",
      "value":"value1"
    }
  ],
  "sys_tags":[]
}
],
"total_count":1
}
```

## 状态码

- 正常

状态码	描述
200	查询资源实例列表成功。

- 异常

状态码	描述
400	标签等参数非法。
401	认证失败。
403	鉴权失败。
404	资源未找到。
500	系统异常。

## 5.3.5 查询资源实例数量 - ListResourceQuantityByTag

### 功能介绍

根据标签查询资源实例数量。

### 授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

### URI

- POST /v3/sfs/tms/{project\_id}/file-systems/resource-instances/count
- 参数说明

参数	是否必选	参数类型	描述
project_id	是	String	操作用户的项目ID，获取方法请参见 <a href="#">获取项目ID</a> 。

请求参数

表 5-34 请求 Header 参数

参数	是否必选	参数类型	描述
Content-type	是	String	发送的实体的MIME类型。 示例：application/json
X-Auth-Token	否	String	用户Token。

表 5-35 请求 Body 参数

参数	是否必选	参数类型	描述
without_any_tag	否	boolean	不包含任意一个标签，该字段为true时查询所有不带标签的资源，此时忽略“tags”字段。该字段为false或者未提供该参数时，该条件不生效，即返回所有资源或按"tags", "matches"等条件过滤。
tags	否	List< <a href="#">tag</a> >	包含标签，最多包含20个key，每个key下面的value最多20个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，同一个key中values不能重复。结果返回包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无tag过滤条件时返回全量数据。 详细说明参见 <a href="#">表5-30</a> 。

参数	是否必选	参数类型	描述
sys_tags	否	List<tag>	包含系统标签。仅op_service权限可以使用此字段。 sys_tags和过滤条件without_any_tag 、tags不能同时使用。 目前此字段只包含一个tag结构体： key： _sys_enterprise_project_id value：企业项目id列表 目前key下面只包含一个value。 0表示默认企业项目。 详细说明参见表5-30。
matches	否	List<match>	搜索字段。match中key为固定字典值，当前key固定为resource_name，表示根据key对应的value值实现前缀搜索，后续再扩展。 详细说明参见表5-31。

响应参数

表 5-36 响应 body 参数

参数	是否必选	参数类型	描述
total_count	是	Integer	总记录数

请求示例

projectId为“c80a2157ba1d46c0825265947342077c”，查询资源实例数量示例：

POST https://{endpoint}/v3/sfs/tms/c80a2157ba1d46c0825265947342077c/file-systems/resource-instances/count

请求体示例：

```
{
  "tags": [
    {
      "key": "key1",
      "values": [
        "value1",
        "value2"
      ]
    },
    {
      "key": "key2",
      "values": [

```

```
        "value1",  
        "value2"  
    ]  
  },  
  ],  
  "matches":[  
    {  
      "key":"resource_name",  
      "value":"resource1"  
    }  
  ],  
  "without_any_tag":"true"  
}
```

## 响应示例

```
{  
  "total_count":1  
}
```

## 状态码

- 正常

状态码	描述
200	查询资源实例数量成功。

- 异常

状态码	描述
400	标签等参数非法
401	认证失败
403	鉴权失败
404	资源未找到
500	系统异常

## 5.3.6 查询项目标签 - ListTagsByProject

### 功能介绍

查询租户在指定Project中所有资源的标签集合。

### 授权信息

账号具备所有API的调用权限，如果使用账号下的IAM用户调用当前API，该IAM用户需具备调用API所需的权限，具体权限要求请参见[权限和授权项](#)。

URI

- GET /v3/sfs/tms/{project\_id}/file-systems/tags
- 参数说明

参数	是否必选	参数类型	描述
project_id	是	String	操作用户的项目ID，获取方法请参见 <a href="#">获取项目ID</a> 。

请求参数

表 5-37 请求 Header 参数

参数	是否必选	参数类型	描述
Content-type	是	String	发送的实体的MIME类型。 示例：application/json。
X-Auth-Token	否	String	用户Token。

响应参数

表 5-38 响应 body 参数

参数	是否必选	参数类型	描述
tags	是	List< <a href="#">tag</a> >	标签列表。详细说明参见 <a href="#">表 5-30</a> 。

请求示例

projectId为 “c80a2157ba1d46c0825265947342077c” ， 查询项目标签示例：

GET https://{endpoint}/v3/sfs/tms/c80a2157ba1d46c0825265947342077c/file-systems/tags

响应示例

```
{
  "tags": [
    {
      "key": "key1",
      "values": [
        "value1",
        "value2"
      ]
    },
    {
      "key": "key2",
      "values": [
        "value1",
        "value2"
      ]
    }
  ]
}
```

```
    ]  
  }  
]  
}
```

状态码

● 正常

状态码	描述
200	查询项目标签成功。

● 异常

状态码	描述
400	标签等参数非法。
401	认证失败。
403	鉴权失败。
404	资源未找到。
500	系统异常。

# 6 权限和授权项

## 6.1 权限及授权项说明

如果您需要对您所拥有的SFS进行精细的权限管理，您可以使用统一身份认证服务（Identity and Access Management，简称IAM），如果华为账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用SFS服务的其它功能。

默认情况下，新建的IAM用户没有任何权限，您需要将其加入用户组，并给用户组授予策略或角色，才能使用户组中的用户获得相应的权限，这一过程称为授权。授权后，用户就可以基于已有权限对云服务进行操作。

权限根据授权的精细程度，分为角色和策略。角色以服务为粒度，是IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。策略以API接口为粒度进行权限拆分，授权更加精细，可以精确到某个操作、资源和条件，能够满足企业对权限最小化的安全管控要求。

### 说明

如果您要允许或是禁止某个接口的操作权限，请使用策略。

账号具备所有接口的调用权限，如果使用账号下的IAM用户发起API请求时，该IAM用户必须具备调用该接口所需的权限，否则，API请求将调用失败。每个接口所需要的权限，与各个接口所对应的授权项相对应，只有发起请求的用户被授予授权项所对应的策略，该用户才能成功调用该接口。例如，用户要调用接口来查询云服务器列表，那么这个IAM用户被授予的策略中必须包含允许“ecs:servers:list”的授权项，该接口才能调用成功。

## 支持的授权项

策略包含系统策略和自定义策略，如果系统策略不满足授权要求，管理员可以创建自定义策略，并通过给用户组授予自定义策略来进行精细的访问控制。策略支持的操作与API相对应，授权项列表说明如下：

- 权限：允许或拒绝某项操作。
- 对应API接口：自定义策略实际调用的API接口。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。

- 依赖的授权项：部分Action存在对其他Action的依赖，需要将依赖的Action同时写入授权项，才能实现对应的权限功能。
- IAM项目(Project)/企业项目(Enterprise Project)：自定义策略的授权范围，包括IAM项目与企业项目。授权范围如果同时支持IAM项目和企业项目，表示此授权项对应的自定义策略，可以在IAM和企业管理两个服务中给用户组授权并生效。如果仅支持IAM项目，不支持企业项目，表示仅能在IAM中给用户组授权并生效，如果在企业管理中授权，则该自定义策略不生效。关于IAM项目与企业项目的区别，详情请参见：[IAM与企业管理的区别](#)。

弹性文件服务支持的自定义策略授权项如下所示：

- 通用文件系统管理**：包含文件系统管理相关接口对应的授权项，如创建文件系统、获取文件系统列表、删除文件系统等接口。
- 访问规则管理**：包含访问规则管理相关接口对应的授权项，如设置文件系统ACL、获取文件系统ACL、删除文件系统ACL等接口。
- 标签管理**：包含标签管理相关接口对应的授权项，如批量添加资源标签、批量删除资源标签、查询资源标签、查询资源实例列表、查询资源实例数量、查询项目标签等接口。

## 6.2 授权项分类

### 说明

“√”表示支持，“x”表示暂不支持。

### 通用文件系统管理

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
创建文件系统	PUT /{file-system-name}	sfs3:fileSystem:createFileSystem	x	√
获取文件系统列表	GET /	sfs3:fileSystem:listFileSystems	x	√
删除文件系统	DELETE /{file-system-name}	sfs3:fileSystem:deleteFileSystem	x	√

访问规则管理

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
设置文件系统ACL	PUT /{file-system-name}?sfsacl	sfs3:fileSystem:putACL	×	√
获取文件系统ACL	GET /{file_system_name}?sfsacl	sfs3:fileSystem:getACL	×	√
删除文件系统ACL	DELETE /{file_system_name}?sfsacl	sfs3:fileSystem:deleteACL	×	√

标签管理

表 6-1 标签管理权限项列表

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
批量添加资源标签	POST /v3/sfs/tms/{project_id}/file-systems/{resource_id}/tags/create	sfs3:tags:tagResource	×	√
批量删除资源标签	POST /v3/sfs/tms/{project_id}/file-systems/{resource_id}/tags/delete	sfs3:tags:unTagResource	×	√
查询资源标签	GET /v3/sfs/tms/{project_id}/file-systems/{resource_id}/tags	sfs3:tags:listTagsForResource	×	√

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询资源实例列表	POST /v3/sfs/tms/{project_id}/file-systems/resource-instances/filter	sfs3:tags:listResourcesByTag	×	×
查询资源实例数量	POST /v3/sfs/tms/{project_id}/file-systems/resource-instances/count	sfs3:tags:listResourcesByTag	×	×
查询项目标签	GET /v3/sfs/tms/{project_id}/file-systems/tags	sfs3:tags:listTags	×	×

# 7 附录

## 7.1 状态码

当您调用API时，如果遇到“APIGW”开头的错误码，请参见[API网关错误码](#)进行处理。

• 正常

返回值	说明
200 OK	GET和PUT操作正常返回。
201 Created	POST操作正常返回。
202 Accepted	请求已被接受。
204 No Content	DELETE操作正常返回。

• 异常

返回值	说明
400 Bad Request	服务器未能处理请求。
401 Unauthorized	被请求的页面需要用户名和密码。
403 Forbidden	对被请求页面的访问被禁止。
404 Not Found	服务器无法找到被请求的页面。
405 Method Not Allowed	请求中指定的方法不被允许。
406 Not Acceptable	服务器生成的响应无法被客户端所接受。
407 Proxy Authentication Required	用户必须首先使用代理服务器进行验证，这样请求才会被处理。
408 Request Timeout	请求超出了服务器的等待时间。
409 Conflict	由于冲突，请求无法被完成。

返回值	说明
500 Internal Server Error	请求未完成。服务异常。
501 Not Implemented	请求未完成。服务器不支持所请求的功能。
502 Bad Gateway	请求未完成。服务器从上游服务器收到一个无效的响应。
503 Service Unavailable	请求未完成。系统暂时异常。
504 Gateway Timeout	网关超时。

## 7.2 错误码

调用接口出错后，将不会返回结果数据。调用方可根据每个接口对应的错误码来定位错误原因。当调用出错时，HTTP请求返回一个3xx，4xx或5xx的HTTP状态码。返回的消息体中是具体的错误代码及错误信息。在调用方找不到错误原因时，可以联系华为云客服，并提供错误码，以便我们尽快帮您解决问题。

### 错误响应消息格式

当错误发生时，响应消息头中都会包含：

- Content-Type: application/xml
- 错误对应的3xx，4xx或5xx的HTTP状态码。

响应消息体中同样会包含对错误的描述信息。下面的错误响应消息体示例展示了所有REST错误响应中公共的元素。

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
<Code>NoSuchKey</Code>
<Message>The resource you requested does not exist</Message>
<Resource>/example-filessystem/object</Resource>
<RequestId>001B21A61C6C0000013402C4616D5285</RequestId>
<HostId>RkRCRDJENDc5MzdGQkQ4OUY3MTI4NTQ3NDk2Mjg0M0FB
QUFBQUFBYmJiYmJiYmJD</HostId>
</Error>
```

各元素的具体含义如表7-1所示。

表 7-1 错误响应消息元素

元素名称	描述
Error	错误响应消息体XML结构中描述错误信息的根节点元素。
Code	错误响应消息体XML中错误响应对应的HTTP消息返回码，具体的错误码请参见表7-2。
Message	错误响应消息体XML中具体错误更全面、详细的英文解释，具体的错误消息请参见表7-2。
RequestId	本次错误请求的请求ID，用于错误定位。

元素名称	描述
HostId	返回该消息的服务端ID。
Resource	该错误相关的文件系统或对象资源。

 说明

许多错误响应包含其他的更丰富的错误信息，建议将所有错误信息记入日志，方便程序员在诊断程序错误时阅读和理解。

错误码说明

在向SFS系统发出请求后，如果遇到错误，会在响应中包含响应的错误码描述错误信息。通用文件系统的错误码如表7-2所示。

表 7-2 错误码

状态码	错误码	错误信息	处理措施
301 Moved Permanently	PermanentRedirect	尝试访问的文件系统必须使用指定的地址，请将以后的请求发送到这个地址。	按照返回的重定向地址发送请求。
301 Moved Permanently	WebsiteRedirect	Website请求缺少bucketName。	携带文件系统名后重试。
307 Moved Temporarily	TemporaryRedirect	临时重定向，当DNS更新时，请求将被重定向到文件系统。	会自动重定向，也可以将请求发送到重定向地址。
400 Bad Request	BadDigest	客户端指定的对象内容的MD5值与系统接收到的内容MD5值不一致。	检查头域中携带的MD5与消息体计算出来的MD5是否一致。
400 Bad Request	BadDomainName	域名不合法。	使用合法的域名。
400 Bad Request	BadRequest	请求参数不合法。	根据返回的错误消息体提示进行修改。
400 Bad Request	IllegalLocationConstraintException	用户未带Location在非默认Region创文件系统。	请求发往默认Region创文件系统或带非默认Region的Location创文件系统。
400 Bad Request	InvalidArgument	无效的参数。	根据返回的错误消息体提示进行修改。
400 Bad Request	InvalidBucket	请求访问的文件系统已不存在。	更换文件系统名。

状态码	错误码	错误信息	处理措施
400 Bad Request	InvalidBucket Name	请求中指定的文件系统名无效，超长或带不允许的特殊字符。	更换文件系统名。
400 Bad Request	InvalidLocationConstraint	创建文件系统时，指定的Location不合法或不存在。	指定正确的Location创文件系统。
400 Bad Request	InvalidPolicyDocument	表单中的内容与策略文档中指定的条件不一致。	根据返回的错误消息体提示修改构造表单的policy重试。
400 Bad Request	InvalidRedirectLocation	无效的重定向地址。	指定正确的地址。
400 Bad Request	InvalidRequest	无效请求。	根据返回的错误消息体提示进行修改。
400 Bad Request	InvalidRequestBody	请求体无效，需要消息体的请求没有上传消息体。	按照正确的格式上传消息体。
400 Bad Request	KeyTooLongError	提供的Key过长。	使用较短的Key。
400 Bad Request	MalformedError	请求中携带的XML格式不正确。	使用正确的XML格式重试。
400 Bad Request	MalformedQuotaError	Quota的XML格式不正确。	使用正确的XML格式重试。
400 Bad Request	MalformedXML	当用户发送了一个配置项的错误格式的XML会出现这样的错误。	使用正确的XML格式重试。
400 Bad Request	MetadataTooLarge	元数据消息头超过了允许的最大元数据大小。	减少元数据消息头。
400 Bad Request	MissingRegion	请求中缺少Region信息，且系统无默认Region。	请求中携带Region信息。
400 Bad Request	MissingRequestBodyError	当用户发送一个空的XML文档作为请求时会发生。	提供正确的XML文档。
400 Bad Request	MissingRequiredHeader	请求中缺少必要的头域。	提供必要的头域。
400 Bad Request	MissingSecurityHeader	请求缺少一个必须的头。	提供必要的头域。

状态码	错误码	错误信息	处理措施
400 Bad Request	TooManyBuckets	用户拥有的文件系统的数量达到了系统的上限，并且请求试图创建一个新文件系统。	删除部分文件系统后重试。
400 Bad Request	TooManyWrongSignature	因高频错误请求被拒绝服务。	更换正确的Access Key后重试。
400 Bad Request	UnexpectedContent	该请求需要消息体而客户端没带，或该请求不需要消息体而客户端带了。	根据说明重试。
400 Bad Request	ContentSHA256Mismatch	客户端与服务端计算的对象的SHA256值不一致。	建议重新确认客户端计算的对象的SHA256的值是否正确。
403 Forbidden	AccessDenied	拒绝访问，请求没有携带日期头域或者头域格式错误。	请求携带正确的日期头域。
403 Forbidden	DeregisterUser	用户已经注销。	充值或重新开户。
403 Forbidden	InArrearOrInsufficientBalance	用户欠费或余额不足而没有权限进行某种操作。	充值。
403 Forbidden	InvalidAccessKeyId	系统记录中不存在客户提供的Access Key Id。	携带正确的Access Key Id。
403 Forbidden	RequestTimeTooSkewed	客户端发起请求的时间与OBS服务端的时间相差太大。 出于安全目的，OBS会校验客户端与OBS服务端的时间差，当该时间差大于15分钟时，OBS服务端会拒绝您的请求，从而出现此报错。	请检查客户端时间是否与当前OBS服务端时间相差太大。请根据本地UTC时间调整客户端时间后再访问。
403 Forbidden	SignatureDoesNotMatch	请求中带的签名与系统计算得到的签名不一致。	检查你的Secret Access Key和签名计算方法。详细排查方法请参见 <a href="#">签名不匹配（SignatureDoesNot Match）如何处理</a> 。
403 Forbidden	VirtualHostDomainRequired	未使用虚拟主机访问域名。	Host使用 <a href="#">虚拟主机访问域名</a> 。
403 Forbidden	Unauthorized	用户未实名认证。	请实名认证后重试。

状态码	错误码	错误信息	处理措施
404 Not Found	NoSuchBucket	指定的文件系统不存在。	先创文件系统再操作。
404 Not Found	NoSuchLifecycleConfiguration	请求的Lifecycle不存在。	先配置Lifecycle。
405 Method Not Allowed	MethodNotAllowed	指定的方法不允许操作在请求的资源上。 对应返回的Message为: Specified method is not supported.	方法不允许。
408 Request Timeout	RequestTimeout	用户与Server之间的socket连接在超时时间内没有进行读写操作。	检查网络后重试，或联系技术支持。
409 Conflict	BucketAlreadyExists	请求的文件系统名已经存在。文件系统的命名空间是系统中所有用户共用的，选择一个不同的文件系统名再重试一次。	更换文件系统名。
409 Conflict	BucketAlreadyOwnedByYou	发起该请求的用户已经创建过了这个名字的文件系统，并拥有这个文件系统。	不需要再创文件系统了。
409 Conflict	BucketNotEmpty	用户尝试删除的文件系统不为空。	先删除文件系统中对象，然后再删文件系统。
409 Conflict	ServiceNotSupported	请求的方法服务端不支持。	服务端不支持，请联系技术支持。
411 Length Required	MissingContentLength	必须要提供HTTP消息头中的Content-Length字段。	提供Content-Length消息头。
412 Precondition Failed	PreconditionFailed	用户指定的先决条件中至少有一项没有包含。	根据返回消息体中的Condition提示进行修改。
500 Internal Server Error	InternalError	系统遇到内部错误，请重试。	请联系技术支持。
501 Not Implemented	ServiceNotImplemented	请求的方法服务端没有实现。	当前不支持，请联系技术支持。
503 Service Unavailable	ServiceUnavailable	服务器过载或者内部错误异常。	等待一段时间后重试，或联系技术支持。

状态码	错误码	错误信息	处理措施
503 Service Unavailable	SlowDown	请降低请求频率。	请降低请求频率。

## 7.3 获取访问密钥（AK/SK）

### 说明

IAM用户若希望使用AK/SK访问SFS，必须要账号为其开启“编程访问”。开启方法请参见[查看或修改IAM用户信息](#)。

在调用接口的时候，需要使用AK/SK进行签名验证。AK/SK获取步骤如下：

- 步骤1** 登录控制台。
- 步骤2** 鼠标指向界面右上角的登录用户名，在下拉列表中单击“我的凭证”。
- 步骤3** 单击“管理访问密钥”。
- 步骤4** 在左侧导航栏单击“访问密钥”。
- 步骤5** 单击“新增访问密钥”，进入“新增访问密钥”页面。
- 步骤6** 输入描述（可选），单击“确定”。
- 步骤7** 通过邮箱、手机或者虚拟MFA进行验证，输入对应的验证码。如果没有开启敏感操作保护，请跳过该步骤。
- 步骤8** 单击“立即下载”，浏览器自动下载访问密钥。

### 说明

为防止访问密钥泄露，建议您将其保存到安全的位置。

----结束

## 7.4 获取项目 ID

### 操作场景

在调用接口的时候，部分URL中需要填入项目ID，所以需要获取到项目ID。有如下两种获取方式：

- [调用API获取项目ID](#)
- [从控制台获取项目ID](#)

### 调用 API 获取项目 ID

项目ID可以通过调用[查询指定条件下的项目列表](#)API获取。

获取项目ID的接口为“GET https://{Endpoint}/v3/projects”，其中{Endpoint}为IAM的终端节点，可以从[地区和终端节点](#)获取。接口的认证鉴权请参见[认证鉴权](#)。

响应示例如下，其中projects下的“id”即为项目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"
  }
}
```

## 从控制台获取项目 ID

从控制台获取项目ID的步骤如下：

1. 登录管理控制台。
2. 鼠标悬停在右上角的用户名，选择下拉列表中的“我的凭证”。  
在“API凭证”页面的项目列表中查看项目ID。

图 7-1 查看项目 ID

