

对象存储服务

Node.js SDK 开发指南

文档版本 01
发布日期 2025-02-07



版权所有 © 华为技术有限公司 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。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 SDK 下载(Node.js SDK)	1
2 示例程序(Node.js SDK)	3
3 快速入门(Node.js SDK)	4
3.1 使用前需知(Node.js SDK).....	4
3.2 OBS 服务环境搭建(Node.js SDK).....	4
3.3 开发环境准备(Node.js SDK).....	7
3.4 安装 SDK(Node.js SDK).....	7
3.5 获取服务地址(Node.js SDK).....	8
3.6 初始化 OBS 客户端(Node.js SDK).....	8
3.7 创建桶(Node.js SDK).....	9
3.8 上传对象(Node.js SDK).....	10
3.9 下载对象(Node.js SDK).....	11
3.10 列举对象(Node.js SDK).....	12
3.11 删除对象(Node.js SDK).....	13
3.12 OBS 客户端通用示例(Node.js SDK).....	14
4 初始化(Node.js SDK)	18
4.1 配置密钥(Node.js SDK).....	18
4.2 创建 OBS 客户端(Node.js SDK).....	18
4.3 配置 OBS 客户端(Node.js SDK).....	20
4.4 配置 SDK 日志(Node.js SDK).....	22
4.5 配置 SDK 代理(Node.js SDK).....	22
5 管理桶(Node.js SDK)	23
5.1 创建桶(Node.js SDK).....	23
5.2 列举桶列表(Node.js SDK).....	33
5.3 判断桶是否存在(Node.js SDK).....	39
5.4 删除桶(Node.js SDK).....	43
5.5 获取桶元数据(Node.js SDK).....	47
5.6 设置桶 ACL(Node.js SDK).....	54
5.7 获取桶 ACL(Node.js SDK).....	65
5.8 设置桶策略(Node.js SDK).....	71
5.9 获取桶策略(Node.js SDK).....	75
5.10 删除桶策略(Node.js SDK).....	79

5.11 获取桶区域位置(Node.js SDK).....	83
5.12 获取桶存量信息(Node.js SDK).....	87
5.13 设置桶配额(Node.js SDK).....	91
5.14 获取桶配额(Node.js SDK).....	95
5.15 设置桶存储类别(Node.js SDK).....	99
5.16 获取桶存储类别(Node.js SDK).....	103
6 上传对象(Node.js SDK).....	108
6.1 对象上传简介(Node.js SDK).....	108
6.2 文本上传(Node.js SDK).....	108
6.3 流式上传(Node.js SDK).....	123
6.4 文件上传(Node.js SDK).....	139
6.5 创建文件夹(Node.js SDK).....	154
6.6 设置对象元数据(Node.js SDK).....	169
6.7 上传对象时设置指定对象生命周期(Node.js SDK).....	181
6.8 追加上传(Node.js SDK).....	183
6.9 断点续传上传(Node.js SDK).....	199
6.10 基于表单上传(Node.js SDK).....	213
7 下载对象(Node.js SDK).....	219
7.1 对象下载简介(Node.js SDK).....	219
7.2 文本下载(Node.js SDK).....	219
7.3 流式下载(Node.js SDK).....	231
7.4 文件下载(Node.js SDK).....	243
7.5 范围下载(Node.js SDK).....	255
7.6 限定条件下载(Node.js SDK).....	267
7.7 重写响应头(Node.js SDK).....	279
7.8 下载归档存储对象(Node.js SDK).....	280
7.9 断点续传下载(Node.js SDK).....	286
7.10 图片处理(Node.js SDK).....	296
8 管理对象(Node.js SDK).....	309
8.1 设置对象元数据(Node.js SDK).....	309
8.2 获取对象元数据(Node.js SDK).....	321
8.3 设置对象 ACL(Node.js SDK).....	332
8.4 获取对象 ACL(Node.js SDK).....	344
8.5 列举桶内对象(Node.js SDK).....	352
8.6 删除对象(Node.js SDK).....	367
8.7 批量删除对象(Node.js SDK).....	372
8.8 复制对象(Node.js SDK).....	380
9 多段相关接口(Node.js SDK).....	400
9.1 多段相关接口说明(Node.js SDK).....	400
9.2 初始化多段上传任务(Node.js SDK).....	403
9.3 上传段(Node.js SDK).....	417

9.4 合并段(Node.js SDK).....	426
9.5 列举已上传的段(Node.js SDK).....	433
9.6 列举分段上传任务(Node.js SDK).....	442
9.7 复制段(Node.js SDK).....	453
9.8 取消分段上传任务(Node.js SDK).....	459
10 临时授权访问(Node.js SDK).....	465
10.1 使用临时 URL 进行授权访问(Node.js SDK).....	465
11 多版本控制(Node.js SDK).....	476
11.1 多版本控制简介(Node.js SDK).....	476
11.2 设置桶多版本状态(Node.js SDK).....	476
11.3 查看桶多版本状态(Node.js SDK).....	480
11.4 获取多版本对象(Node.js SDK).....	484
11.5 复制多版本对象(Node.js SDK).....	496
11.6 恢复多版本归档存储对象(Node.js SDK).....	511
11.7 列举桶内多版本对象(Node.js SDK).....	517
11.8 设置多版本对象权限(Node.js SDK).....	534
11.9 获取多版本对象权限(Node.js SDK).....	535
11.10 删除多版本对象(Node.js SDK).....	536
12 生命周期管理(Node.js SDK).....	539
12.1 设置桶的生命周期配置(Node.js SDK).....	539
12.2 获取桶的生命周期配置(Node.js SDK).....	550
12.3 删除桶的生命周期配置(Node.js SDK).....	557
13 桶的 CORS 配置(Node.js SDK).....	561
13.1 设置桶的 CORS 配置(Node.js SDK).....	561
13.2 获取桶的 CORS 配置(Node.js SDK).....	567
13.3 删除桶的 CORS 配置(Node.js SDK).....	573
14 设置访问日志(Node.js SDK).....	577
14.1 设置桶日志管理配置(Node.js SDK).....	577
14.2 获取桶日志管理配置(Node.js SDK).....	587
15 静态网站托管(Node.js SDK).....	594
15.1 静态网站托管简介(Node.js SDK).....	594
15.2 设置托管配置(Node.js SDK).....	595
15.3 获取托管配置(Node.js SDK).....	605
15.4 删除托管配置(Node.js SDK).....	612
16 标签管理(Node.js SDK).....	616
16.1 设置桶标签(Node.js SDK).....	616
16.2 获取桶标签(Node.js SDK).....	621
16.3 删除桶标签(Node.js SDK).....	625
17 服务端加密(Node.js SDK).....	630

17.1 服务端加密简介(Node.js SDK).....	630
17.2 加密说明(Node.js SDK).....	630
17.3 加密示例(Node.js SDK).....	632
18 异常处理(Node.js SDK).....	634
18.1 OBS 服务端错误码(Node.js SDK).....	634
18.2 SDK 公共结果对象(Node.js SDK).....	640
18.3 日志分析(Node.js SDK).....	641
18.4 缺少模块异常(Node.js SDK).....	642
18.5 连接超时异常(Node.js SDK).....	642
18.6 签名不匹配异常(Node.js SDK).....	643
19 常见问题(Node.js SDK).....	644
19.1 如何指定 Content-SHA256?(Node.js SDK).....	644
19.2 为什么 SDK 源码中包含 acs.amazonaws.com 关键字? (Node.js SDK).....	645
19.3 如何获取账号 ID 和用户 ID? (Node.js SDK).....	645
19.4 如何理解 Content-Type (MIME) (Node.js SDK).....	647

1 SDK 下载(Node.js SDK)

下载地址

- OBS Node.js SDK最新版本源码：[最新源码下载](#)
- OBS Node.js SDK历史版本下载地址：[历史版本下载](#)

兼容性

- 推荐使用的Node.js版本：Node 0.12.x, Node4.x, Node6.x, Node8.x, Node10.x。
- 接口函数：与旧版本（2.1.x）**不完全兼容**，接口变化如下表：

接口函数	变化说明
ObsClient.listBuckets	响应结果中InterfaceResult.Buckets字段变为数组，去掉InterfaceResult.Buckets.Bucket字段。
ObsClient.setBucketACL	请求参数中Grants字段变为数组，去掉Grants.Grant字段。
ObsClient.getBucketACL	响应结果中InterfaceResult.Grants字段变为数组，去掉InterfaceResult.Grants.Grant字段。
ObsClient.setObjectACL	请求参数中Grants字段变为数组，去掉Grants.Grant字段。
ObsClient.getObjectACL	响应结果中InterfaceResult.Grants字段变为数组，去掉InterfaceResult.Grants.Grant字段。
ObsClient.setBucketLogging	请求参数中LoggingEnabled.TargetGrants字段变为数组，去掉LoggingEnabled.TargetGrants.Grant字段。
ObsClient.getBucketLogging	响应结果中InterfaceResult.LoggingEnabled.TargetGrants字段变为数组，去掉InterfaceResult.LoggingEnabled.TargetGrants.Grant字段。
ObsClient.setBucketWebsite	请求参数中RoutingRules字段变为数组，去掉RoutingRules.RoutingRule字段。

接口函数	变化说明
ObsClient.getBucketWebsite	响应结果中InterfaceResult.RoutingRules字段变为数组，去掉InterfaceResult.RoutingRules.RoutingRule字段。
ObsClient.setBucketCors	请求参数中CorsRule字段改名为CorsRules。
ObsClient.getBucketCors	响应结果中InterfaceResult.CorsRule字段改名为InterfaceResult.CorsRules。
ObsClient.setBucketTagging	请求参数中TagSet字段改名为Tags并变为数组，去掉TagSet.Tag字段。
ObsClient.getBucketTagging	响应结果中InterfaceResult.TagSet字段改名为Tags并变为数组，去掉InterfaceResult.TagSet.Tag字段。

2 示例程序(Node.js SDK)

OBS Node.js SDK提供了丰富的示例程序，方便用户参考或直接使用。您可以从OBS Node.js SDK开发包中获取示例程序，如 `eSDK_Storage_OBS_<VersionId>_Node.js.zip`，解压后 `eSDK_Storage_OBS_<VersionId>_Node.js/examples`为示例程序。您也可以从下面表格中直接下载示例程序。

示例包括以下内容：

代码示例	说明
bucket-operations-sample	展示了桶相关接口的用法
object-operations-sample	展示了对象相关接口的用法
download-sample	展示了下载对象的用法
create-folder-sample	展示了创建文件夹的用法
delete-objects-sample	展示了批量删除对象的用法
list-objects-sample	展示了列举对象的用法
list-versions-sample	展示了列举多版本对象的用法
list-objects-infolder-sample	展示了列举文件夹内对象的用法
object-meta-sample	展示了自定义对象元数据的用法
simple-multipart-upload-sample	展示了分段上传的基本用法
restore-object-sample	展示了下载归档存储对象的用法
concurrent-copypart-sample	展示了分段并发复制大对象的用法
concurrent-download-object-sample	展示了分段并发下载大对象的用法
concurrent-uploadpart-sample	展示了分段并发上传大对象的用法
post-object-sample	展示了表单上传对象的用法
temporary-signature-sample	展示了使用URL进行授权访问的用法

3 快速入门(Node.js SDK)

3.1 使用前需知(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

- 请确认您已经熟悉OBS的基本概念，如**桶 (Bucket)**、**对象 (Object)**、**访问密钥 (AK和SK)** 等。
- 您可以先参考**OBS客户端通用示例**，了解OBS Node.js SDK接口调用的通用方式。
- OBS客户端支持回调函数和Promise对象两种方式返回调用结果。
- 当前各区域特性开放不一致，部分特性只在部分区域开放，使用过程中如果接口HTTP状态码为405，请确认该区域是否支持该功能特性。

3.2 OBS 服务环境搭建(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

步骤1 注册云服务账号

使用OBS之前必须要有一个云服务账号。

1. 打开浏览器。
2. 登录[公有云网站](#)。
3. 在页面右上角单击“注册”。
4. 按需填写注册信息并单击“同意协议并注册”。

步骤2 开通OBS服务

使用OBS服务之前必须先充值，才能正常使用OBS服务。

1. 登录[管理控制台](#)。
2. 单击页面右上角的“费用和成本”进入费用中心页面。
3. 选择“资金管理 > 充值”，系统自动跳转到充值窗口。
4. 根据界面提示信息，对账户进行充值。
5. 充值成功后，关闭充值窗口，返回管理控制台首页。
6. 在服务列表中选择“对象存储服务 OBS”，开通并进入OBS管理控制台。

步骤3 创建访问密钥

OBS通过用户账户中的AK和SK进行签名验证，确保通过授权的账户才能访问指定的OBS资源。以下是对AK和SK的解释说明：

- AK: Access Key ID，接入键标识，用户在对象存储服务系统中的接入键标识，一个接入键标识唯一对应一个用户，一个用户可以同时拥有多个接入键标识。对象存储服务系统通过接入键标识识别访问系统的用户。
- SK: Secret Access Key，安全接入键，用户在对象存储服务系统中的安全接入键，是用户访问对象存储服务系统的密钥，用户根据安全接入键和请求头域生成鉴权信息。安全接入键和接入键标识一一对应。

访问密钥分永久访问密钥（AK/SK）和临时访问密钥（AK/SK和SecurityToken）两种。每个用户最多可创建两个有效的永久访问密钥。临时访问密钥只在设置的有效期内能够访问OBS，过期后需要重新获取。出于安全性考虑，建议您使用临时访问密钥访问OBS，或使用永久访问密钥访问OBS时，定期更新您的访问密钥（AK/SK）。两种密钥的获取方式如下所示。

- 永久访问密钥：
 - a. 登录[管理控制台](#)。
 - b. 单击页面右上角的用户名，并选择“我的凭证”。
 - c. 在“我的凭证”页面，单击左侧导航栏的“访问密钥”。
 - d. 在“访问密钥”页面，单击“新增访问密钥”。



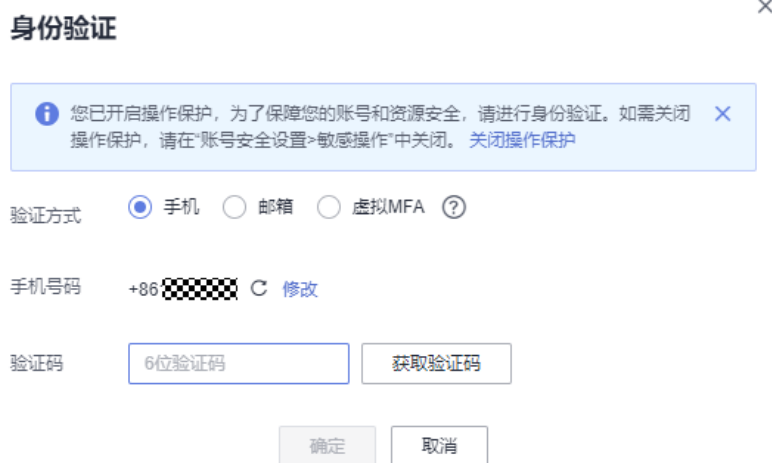
说明

每个用户最多可创建两个有效的访问密钥。

- e. 在弹出的“新增访问密钥”对话框中，输入描述内容（建议），单击“确定”。



- f. (可选) 在弹出的“身份验证”对话框中，选择合适的验证方式进行验证，单击“确定”。



- g. 在弹出的“创建成功”提示框中，单击“立即下载”后，密钥会直接保存到浏览器默认的下载文件夹中。



- h. 打开下载下来的“credentials.csv”文件即可获得到访问密钥（AK和SK）。

📖 说明

- 在密钥文件中，Access Key ID列对应的值即AK，Secret Access Key列对应的值即SK。
 - 为防止访问密钥泄露，建议您将其保存到安全的位置。如果用户在此提示框中单击“取消”，则不会下载密钥，后续也将无法重新下载。如果需要使用访问密钥，可以重新创建新的访问密钥。
- 临时访问密钥：
临时AK/SK和SecurityToken是系统颁发给用户的临时访问令牌，通过接口设置有效期，范围为15分钟至24小时，过期后需要重新获取。临时AK/SK和SecurityToken遵循权限最小化原则。使用临时AK/SK鉴权时，临时AK/SK和SecurityToken必须同时使用。

获取临时访问密钥的接口请参考[获取临时AK/SK和securitytoken](#)。

须知

OBS属于全局级服务，所以在获取临时访问密钥时，需要设置Token的使用范围取值为domain，表示获取的Token可以作用于全局服务，全局服务不区分项目或者区域。

----结束

3.3 开发环境准备(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

- 从[Node.js官网](#)下载并安装推荐使用的版本。
- 从[Eclipse官网](#)下载并安装Eclipse IDE for JavaScript and Web Developer最新版本。

3.4 安装 SDK(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

使用 npm 安装（推荐）

步骤1 运行`npm -v`命令查看npm版本并确保npm已安装。

步骤2 运行`npm install esdk-obs-nodejs`命令执行安装。

----结束

📖 说明

- 如果您使用的是Windows操作系统，当运行npm命令时提示“不是内部或外部命令”，请在Path环境变量中增加npm的安装目录（一般为Node.js的安装目录）。
- 您可能需要重启电脑使环境变量生效。
- 如果您使用npm安装依赖时出现网络错误，请使用代理。

使用源码安装

以安装OBS Node.js SDK最新版本为例，步骤如下：

步骤1 下载OBS Node.js SDK开发包。

步骤2 解压该开发包，可以看到其中包含examples文件夹（代码示例），lib文件夹（SDK源码），package.json文件（依赖配置文件）和README.txt（SDK版本特性描述文件）。

步骤3 命令行切换到SDK开发包解压目录，运行`npm install`安装依赖库，生成node_modules文件夹。

步骤4 （可选）在Eclipse JavaScript项目中导入源码。打开Eclipse JavaScript IDE，选择Import > Projects from Folder or Archive，并在Import source中选择SDK开发包解压目录。

----结束

📖 说明

安装完成后，您的目录结构应该像下面这样：

```
├── examples
├── lib
├── node_modules
├── package.json
└── README.txt
```

3.5 获取服务地址(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

- 您可以从[这里](#)查看OBS当前开通的服务地址和区域信息。

须知

SDK支持带协议名和不带协议名两种方式传入服务地址，例如获取到的服务地址为“your-endpoint”，则初始化OBS客户端时传入的服务地址可以为“http://your-endpoint”、“https://your-endpoint”和“your-endpoint”三种形式。

3.6 初始化 OBS 客户端(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

向OBS发送任一HTTP/HTTPS请求之前，必须先创建一个ObsClient实例：

```
// 引入obs库
// 使用npm安装
```

```
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// 使用访问OBS

// 关闭obsClient
// obsClient.close();
```

注意

- 由于JavaScript是异步编程语言, 所以不能在访问OBS期间调用close方法。
- obsClient在调用obsClient.close方法关闭后不能再次使用。

说明

- 更多关于OBS客户端初始化的内容请参考“初始化”章节。
- 日志配置详见[配置SDK日志](#)。
- 代理配置详见[配置SDK代理\(Node.js SDK\)](#)。

3.7 创建桶(Node.js SDK)

须知

开发过程中, 您有任何问题可以在github上[提交issue](#)。

桶是OBS全局命名空间, 相当于数据的容器、文件系统的根目录, 可以存储若干对象。以下代码展示如何新建一个桶:

📖 说明

- 桶的名字是全局唯一的，所以您需要确保不与已有的桶名称重复。
- 桶命名规则如下：
 - 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。
 - 禁止使用类IP地址。
 - 禁止以“-”或“.”开头及结尾。
 - 禁止两个“.”相邻（如：“my..bucket”）。
 - 禁止“-”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。
- 同一用户多次创建同名桶不会报错，创建的桶属性以第一次请求为准。
- 更多创建桶的信息，请参见[创建桶](#)。

3.8 上传对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

本示例用于上传字符串“Hello OBS”到桶名为“examplebucket”里，名称为“example/objectname”。

代码示例如下所示：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AK/SK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定文本对象
      Body: 'Hello OBS'
    };
    // 文本上传对象
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put object(%s) under the bucket(%s) successful!!!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
    }
  }
}
```

```
console.log("StorageClass:%s, ETag:%s", result.InterfaceResult.StorageClass, result.InterfaceResult.ETag);
return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
console.log(error);
};
};
};
putObject();
```

📖 说明

- 更多上传对象的信息，请参见[上传对象](#)。

3.9 下载对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

本示例用于下载桶名为“examplebucket”里，名称为“example/objectname”的对象。

代码示例如下所示：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');
// 创建ObsClient实例
const obsClient = new ObsClient({
// 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
// security_token: process.env.SECURITY_TOKEN,
// endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
try {
const params = {
// 指定存储桶名称
Bucket: "examplebucket",
// 指定对象名，此处以 example/objectname 为例
Key: 'example/objectname',
};

// 文本下载对象
const result = await obsClient.getObject(params);
```

```
if (result.CommonMsg.Status <= 300) {
  console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  console.log('Object Content: %s', result.InterfaceResult.Content);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};
getObject();
```

📖 说明

- 更多下载对象的信息，请参见[下载对象](#)。

3.10 列举对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

当完成一系列上传对象操作后，可能需要查看桶中包含哪些对象。以下代码展示如何列举指定桶中的对象：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AK/SK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
    };
    // 列举桶内对象
    const result = await obsClient.listObjects(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
    }
  }
}
```

```
console.log("RequestId: %s", result.CommonMsg.RequestId);
for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
  const val = result.InterfaceResult.Contents[j];
  console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
    j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
};
return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
listObjects();
```

📖 说明

- 上面的代码默认列举1000个对象（Object）。
- 更丰富的列举功能，请参见[列举对象](#)。

3.11 删除对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

本示例用于删除桶名为“examplebucket”里，名称为“example/objectname”的对象。

代码示例如下所示：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteObject() {
  try {
    const params = {
```

```
// 指定存储桶名称
Bucket: "examplebucket",
// 指定删除对象，此处以 example/objectname 为例
Key: 'example/objectname',
};
// 删除对象
const result = await obsClient.deleteObject(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Delete object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
deleteObject();
```

📖 说明

- 更多删除对象的信息，请参见[删除对象](#)。

3.12 OBS 客户端通用示例(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

回调函数返回结果：

OBS客户端可通过回调函数的形式返回结果，回调函数依次包含异常信息和**SDK公共结果对象**两个参数。如果回调函数中异常信息参数不为空，则表明接口调用异常；反之，则表明接口调用完成，此时应从**SDK公共结果对象**中获取HTTP状态码，判断操作是否成功。代码示例如下：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
```

```
});  
  
// 构造桶操作请求参数  
var requestParam1 = {  
    Bucket : 'bucketname'  
    // 其他字段  
};  
  
var callback1 = (err, result) => {  
    // 处理桶操作调用结果  
};  
  
// 调用桶操作接口，如创建桶  
obsClient.createBucket(requestParam1, callback1);  
  
//构造对象操作请求参数  
var requestParam2 = {  
    Bucket : 'bucketname',  
    Key : 'objectname'  
    // 其他字段  
};  
  
var callback2 = (err, result) => {  
    // 处理对象操作调用结果  
};  
  
// 调用对象操作接口，如下载对象  
obsClient.getObject(requestParam2, callback2);
```

说明

对于桶操作接口，请求对象中固定包含Bucket字段用于指定桶名；对于对象操作接口，请求对象中固定包含Bucket字段和Key字段分别用于指定桶名与对象名。

以下代码展示了使用回调函数返回调用结果的通用示例：

```
// 引入obs库  
// 使用npm安装  
const ObsClient = require("esdk-obs-nodejs");  
// 使用源码安装  
// var ObsClient = require('./lib/obs');  
  
// 创建ObsClient实例  
const obsClient = new ObsClient({  
    // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html  
    access_key_id: process.env.ACCESS_KEY_ID,  
    secret_access_key: process.env.SECRET_ACCESS_KEY,  
    // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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  
    // security_token: process.env.SECURITY_TOKEN,  
    // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写  
    server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
// 调用接口进行操作，例如上传对象  
obsClient.putObject({  
    Bucket : 'bucketname',  
    Key : 'objectname',  
    Body : 'Hello OBS'  
}, (err, result) => {  
    // 异常信息不为空，表明接口调用异常  
    if(err){  
        console.log('Error-->' + err);  
    }else{  
        // 异常信息为空，表明接口调用完成，应进一步判断HTTP状态码  
        if(result.CommonMsg.Status < 300){// 操作成功  
            if(result.InterfaceResult){
```

```
        // 处理操作成功后业务逻辑
    };
} else { // 操作失败, 获取详细异常信息
    console.log('Code-->' + result.CommonMsg.Code);
    console.log('Message-->' + result.CommonMsg.Message);
    console.log('HostId-->' + result.CommonMsg.HostId);
    console.log('RequestId-->' + result.CommonMsg.RequestId);
}
};
});
```

Promise 对象返回结果:

OBS客户端可通过Promise对象返回结果, 如果通过Promise对象的catch方法没有捕获到异常, 则表明接口调用完成, 此时应从**SDK公共结果对象**中获取HTTP状态码, 判断操作是否成功。代码示例如下:

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
    // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
    access_key_id: process.env.ACCESS_KEY_ID,
    secret_access_key: process.env.SECRET_ACCESS_KEY,
    // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
    // security_token: process.env.SECURITY_TOKEN,
    // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
    server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// 构造桶操作请求参数
var requestParam1 = {
    Bucket: 'bucketname'
    // 其他字段
};

// 调用桶操作接口, 如创建桶
var promise1 = obsClient.createBucket(requestParam1);
promise1.then((result) => {
    // 处理桶操作调用结果
}).catch((err) => {
    // 处理错误
});

// 构造对象操作请求参数
var requestParam2 = {
    Bucket: 'bucketname',
    Key: 'objectname'
    // 其他字段
};

// 调用对象操作接口, 如下载对象
var promise2 = obsClient.getObject(requestParam2);
promise2.then((result) => {
    // 处理对象操作调用结果
}).catch((err) => {
    // 处理错误
});
```

📖 说明

对于桶操作接口，请求对象中固定包含Bucket字段用于指定桶名；对于对象操作接口，请求对象中固定包含Bucket字段和Key字段分别用于指定桶名与对象名。

以下代码展示了使用Promise对象返回调用结果的通用示例：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// 调用接口进行操作，例如上传对象
obsClient.putObject({
  Bucket: 'bucketname',
  Key: 'objectname',
  Body: 'Hello OBS'
}).then((result) => {
  // 接口调用完成，应进一步判断HTTP状态码
  if(result.CommonMsg.Status < 300){// 操作成功
    if(result.InterfaceResult){
      // 处理操作成功后业务逻辑
    };
  }else{// 操作失败，获取详细异常信息
    console.log('Code-->' + result.CommonMsg.Code);
    console.log('Message-->' + result.CommonMsg.Message);
    console.log('HostId-->' + result.CommonMsg.HostId);
    console.log('RequestId-->' + result.CommonMsg.RequestId);
  };
}).catch((err) => {
  // 接口调用发生异常
  console.error('Error-->' + err);
});
```


4 初始化(Node.js SDK)

4.1 配置密钥(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

要接入OBS服务，您需要拥有一组有效的访问密钥（AK和SK）用来进行签名认证。具体可参考[OBS服务环境搭建](#)。

获取AK和SK之后，可以通过创建OBS客户端，调用SDK接口。

获取AK和SK之后，您便可以按照以下步骤进行初始化。

- [创建OBS客户端\(Node.js SDK\)](#)
- [配置OBS客户端\(Node.js SDK\)](#)
- [配置SDK日志\(Node.js SDK\)](#)
- [配置SDK代理\(Node.js SDK\)](#)

4.2 创建 OBS 客户端(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

OBS客户端（ObsClient）是访问OBS服务的Node.js客户端，它为调用者提供一系列与OBS服务进行交互的接口，用于管理、操作桶（Bucket）和对象（Object）等OBS服务上的资源。使用OBS Node.js SDK发起OBS请求，您需要初始化一个ObsClient实例，并根据需要修改客户端初始化配置参数。

通过构造函数创建

- 永久访问密钥（AK/SK）创建OBS客户端代码如下：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// 使用访问OBS

// 关闭obsClient
// obsClient.close();
```

- 临时访问密钥（AK/SK/SecurityToken）创建OBS客户端代码如下：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// 使用访问OBS

// 关闭obsClient
// obsClient.close();
```

通过工厂方法创建

- 永久访问密钥（AK/SK）创建OBS客户端代码如下：

```
// 引入obs库
// 使用npm安装
var ObsClient = require('esdk-obs-nodejs');
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 通过工厂方法初始化ObsClient实例
var obsClient = new ObsClient();
obsClient.factory({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});
```

```
access_key_id: process.env.ACCESS_KEY_ID,  
secret_access_key: process.env.SECRET_ACCESS_KEY,  
// endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写  
server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
// 使用访问OBS  
  
// 关闭obsClient  
// obsClient.close();
```

- 临时访问密钥 (AK/SK/SecurityToken) 创建OBS客户端代码如下:

```
// 引入obs库  
// 使用npm安装  
var ObsClient = require('esdk-obs-nodejs');  
// 使用源码安装  
// var ObsClient = require('./lib/obs');  
  
// 通过工厂方法初始化ObsClient实例  
var obsClient = new ObsClient();  
obsClient.factory({  
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html  
  access_key_id: process.env.ACCESS_KEY_ID,  
  secret_access_key: process.env.SECRET_ACCESS_KEY,  
  // 如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。  
  // 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  
  security_token: process.env.SECURITY_TOKEN,  
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写  
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
// 使用访问OBS  
  
// 关闭obsClient  
// obsClient.close();
```

📖 说明

- 您的工程中可以有多个obsClient实例, 也可以只有一个。
- obsClient实例在调用close方法关闭后不能再次使用。
- 配置SDK代理请参见[配置SDK代理\(Node.js SDK\)](#)。

4.3 配置 OBS 客户端(Node.js SDK)

须知

开发过程中, 您有任何问题可以在github上[提交issue](#)。

您可通过初始化参数对ObsClient进行配置, 可以配置的参数见下表:

参数	描述	建议值
access_key_id	访问密钥中的AK。	N/A
secret_access_key	访问密钥中的SK。	N/A

参数	描述	建议值
server	连接OBS的服务地址。可包含协议类型、域名、端口号。示例：https://your-endpoint:443。（出于安全性考虑，建议使用https协议）	N/A
max_retry_count	HTTP/HTTPS连接异常时的请求重试次数。默认为3次。	[1, 5]
timeout	HTTP/HTTPS请求超时时间（单位：秒）。默认为60秒。	[10, 60]
http_agent	配置使用http代理，可选项。默认不配置	N/A
https_agent	配置使用https代理，可选项。默认不配置	N/A
ssl_verify	验证服务端证书参数。可能的取值： <ul style="list-style-type: none">• 服务端pem格式根证书文件路径；• true：使用默认的CAs验证服务端证书；• false：表示不验证服务端证书。 默认为false。	N/A
long_conn_param	长连接模式参数（单位：秒）。当该参数大于等于0时，开启长连接模式，并将该参数作为TCP Keep-Alive数据包的 初始延迟 。 默认为空，代表关闭长连接模式。	N/A
is_cname	是否通过自定义域名访问OBS服务。默认为false。	N/A

📖 说明

- 建议值为N/A的表示需要根据实际情况进行设置。
- 如网络状况不佳，建议增大timeout的值。
- 如果设置的server不带协议类型，则默认使用HTTPS协议。

须知

- 如果启用了长连接模式，使用完OBS客户端后必须调用ObsClient.close方法显式关闭，回收连接资源。
- 出于DNS解析性能和OBS服务可靠性的考虑，不允许将server设置为IP，必须使用域名访问OBS服务。
- 配置SDK代理请参见[配置SDK代理\(Node.js SDK\)](#)。

4.4 配置 SDK 日志(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

OBS Node.js SDK基于log4js开源库提供了日志功能，您可以通过ObsClient.initLog开启日志功能并进行配置。代码示例如下：

```
obsClient.initLog({
  file_full_path: './logs/OBS-SDK.log', // 配置日志文件路径
  max_log_size: 20480, // 配置日志文件大小，单位：字节
  backups: 10, // 配置最大可保留的日志文件个数
  level: 'warn', // 配置日志级别
  log_to_console: true // 配置是否将日志打印到console
});
```

说明

- 日志功能默认是关闭的，需要主动开启。
- 您可以从[日志分析](#)章节获取更多关于SDK日志的信息。

4.5 配置 SDK 代理(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

步骤1 在项目工程根目录下使用npm命令下载[proxy-agent](#)依赖包。

```
npm install --save proxy-agent
```

步骤2 在项目工程里面声明并配置代理。

```
const ProxyAgent = require("proxy-agent");
const proxyAgent = new ProxyAgent(`http://username:password@proxyhost:proxyPort`);
```

步骤3 在项目工程里面初始化OBS客户端并配置代理。

```
var obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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
  // 您可以通过登录访问管理控制台获取访问密钥AK/SK，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
  https_agent: proxyAgent,
  http_agent: proxyAgent,
});
```

---结束

5 管理桶(Node.js SDK)

5.1 创建桶(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

OBS桶是对象的容器，您上传的文件都将以对象的形式存放在桶中。本文介绍如何使用Node.js SDK创建桶。

调用创建桶接口，将在当前账号按照用户指定的桶名创建一个新桶，接口支持根据用户诉求，在创建桶的同时配置桶的存储类别、区域及桶的访问权限等参数。

接口约束

- 您必须拥有obs:bucket:CreateBucket权限，才能创建桶。建议使用IAM进行授权，配置方式详见[使用IAM自定义策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。创建桶时，如果使用的终端节点归属于默认区域华北-北京一（cn-north-1），则可以不指定区域；如果使用的终端节点归属于其他区域，则必须指定区域，且指定的区域必须与终端节点归属的区域一致。当前有效的区域名称可从[这里](#)查询。比如初始化时使用的终端节点EndPoint是obs.ap-southeast-1.myhuaweicloud.com，那么在创建桶的时候必须指定Location: ap-southeast-1才会创建成功，否则会返回状态码400。
- 同一账号下，可以创建多个存储桶，数量上限是100个（不区分地域），存储桶中的对象数量和大小没有限制。
- 新创建桶的桶名在OBS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桶时返回HTTP状态码200。除此以外的其他场景重复创建同名桶返回HTTP状态码409，表明桶已存在。
- 用户删除桶后，需要等待30分钟才能创建同名桶和并行文件系统。
- 并不是所有区域都支持创建多AZ桶，您可以在[产品价格详情](#)页面，查询指定区域是否支持多AZ。

方法定义

ObsClient.createBucket(params)

请求参数说明

表 5-1 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>
Location	string	如果请求的OBS服务地址所在区域为默认区域，则可为空，否则为必选。	<p>参数解释: 桶所在的区域。</p> <p>约束限制: 该参数定义了桶将会被创建在哪个区域，如果使用的终端节点是obs.myhuaweicloud.com，可以不携带此参数；如果使用的终端节点不是obs.myhuaweicloud.com，则必须携带此参数。</p> <p>取值范围: 当前有效的OBS区域和终端节点的更多信息，请参考地区和终端节点。终端节点即调用API的请求地址，不同服务不同区域的终端节点不同，您可以向企业管理员获取区域和终端节点信息。</p> <p>默认取值: 终端节点为obs.myhuaweicloud.com且用户未设定区域时，默认为华北-北京一（cn-north-1）。</p>

参数名称	参数类型	是否必选	描述
ACL	AclType	可选	<p>参数解释: 创桶时可指定的预定义访问策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见ACL功能介绍。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的访问策略选项参见AclType。</p> <p>默认取值: private，即私有权限</p>
Storage Class	StorageClassType	可选	<p>参数解释: 创桶时可指定的桶的存储类别。</p> <p>约束限制: 无</p> <p>取值范围: 桶的存储类别参见StorageClassType。</p> <p>默认取值: STANDARD，即标准存储类别</p>
GrantRead	string	可选	<p>参数解释: 被授予READ权限的账号的账号ID，即domain_id。账号被授予READ权限后，针对当前正在创建的桶，允许列举桶内对象、列举桶中多段任务、列举桶中多版本对象、获取桶元数据。</p> <p>约束限制: 无</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID?(Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantWrite	string	可选	<p>参数解释: 被授予WRITE权限的账号的账号ID, 即domain_id。账号被授予WRITE权限后, 针对当前正在创建的桶, 允许创建、删除、覆盖桶内所有对象, 允许初始化段、上传段、复制段、合并段、取消多段上传任务。</p> <p>约束限制: 无</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantReadACP	string	可选	<p>参数解释: 被授予READ_ACP权限的账号的账号ID, 即domain_id。账号被授予READ_ACP权限后, 针对当前正在创建的桶, 允许读桶的ACL信息。</p> <p>约束限制: 无</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantWriteACP	string	可选	<p>参数解释: 被授予WRITE_ACP权限的账号的账号ID, 即domain_id。账号被授予WRITE_ACP权限后, 针对当前正在创建的桶, 允许修改桶的ACL信息。</p> <p>约束限制: 无</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantFullControl	string	可选	<p>参数解释: 被授予FULL_CONTROL权限的账号的账号ID, 即domain_id。账号被授予FULL_CONTROL权限后, 针对当前正在创建的桶, 允许进行任何操作。</p> <p>约束限制: 无</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID?(Node.js SDK)</p> <p>默认取值: 无</p>
GrantReadDelivered	string	可选	<p>参数解释: 被授予READ权限的账号的账号ID, 即domain_id。授权指定账号有READ权限, 并且在默认情况下, 该READ权限将传递给桶内所有对象。</p> <p>约束限制: 无</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID?(Node.js SDK)</p> <p>默认取值: 无</p>
GrantFullControlDelivered	string	可选	<p>参数解释: 被授予FULL_CONTROL权限的账号的账号ID, 即domain_id。授权指定账号有当前桶的全部控制权限, 并且在默认情况下, 该FULL_CONTROL权限将传递给桶内所有对象。</p> <p>约束限制: 无</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID?(Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
MultiEnterprise	string	可选	<p>参数解释: 创桶时可指定的企业项目ID, 开通企业项目的用户可以从企业项目服务获取。</p> <p>约束限制: Epid格式为uuid, 未开通企业项目的用户可以不带该头域。</p> <p>示例: 9892d768-2d13-450f-aac7-ed0e44c2585f</p> <p>默认取值: 无</p>
AzRedundancy	string	可选	<p>参数解释: 创桶时可指定的数据冗余类型, 即AZ类型。</p> <p>约束限制: 归档存储不支持多AZ。如果桶所在区域不支持多AZ存储, 则该桶的存储类别默认为单AZ。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • 如果将桶配置为多AZ, 则取值为3az。 • 如果将桶配置为单AZ, 则无需选择此选项, 系统默认赋值。 <p>默认取值: 单AZ</p>
FileInterface	string	可选	<p>参数解释: 是否创建并行文件系统。</p> <p>约束限制: 无</p> <p>取值范围: 可选值为Enabled</p> <p>默认取值: 无, 不设置则创建对象桶。</p>

表 5-2 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限, 其他任何人都没有访问权限。

常量名	原始值	说明
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	桶公共读，桶内对象公共读。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadDelivered不能应用于对象。
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	桶公共读写，桶内对象公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadWriteDelivered不能应用于对象。

常量名	原始值	说明
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	<p>设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。</p> <p>默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。</p>

表 5-3 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。</p>
ObsClient.enums.StorageClassWarm	WARM	<p>低频访问存储。</p> <p>低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。</p>
ObsClient.enums.StorageClassCold	COLD	<p>归档存储。</p> <p>归档存储适用于很少访问（平均一年访问一次）数据的业务场景。</p>

返回结果说明

表 5-4 返回结果

参数类型	描述
<p>表5-5</p> <p>说明</p> <p>该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释：</p> <p>接口返回信息，详见表5-5。</p>

表 5-5 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	BaseResponseOutput	参数解释: 操作成功后的结果数据，详见 BaseResponseOutput 。 约束限制: 当Status大于300时为空。

表 5-6 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-7 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

本示例用于创建名为examplebucket的桶，并设置所在区域在中国-香港（ap-southeast-1），桶的权限访问控制策略是私有桶，存储类型是低频访问存储，多AZ方式存储。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function createBucket() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定存储桶所在区域，此处以“ap-southeast-1”为例，必须跟传入的Endpoint中Region保持一致
      // Location: "ap-southeast-1",
      // 指定存储桶的权限控制策略，此处以私有读写obs.AclPrivate为例
      ACL: obsClient.enums.AclPrivate,
      // 指定存储桶的存储类型，此处以低频访问存储obs.StorageClassWarm为例。如果未指定该参数，则创建的桶为标准存储
      StorageClass: obsClient.enums.StorageClassWarm,
      // 指定存储桶的AZ类型，此处以“3AZ”为例。不携带时默认为单AZ，如果对应region不支持多AZ存储，则该桶的存储类型仍为单AZ
      AzRedundancy: "3az",
    };
    // 创建桶
    const result = await obsClient.createBucket(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Create bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
```

```
attempting to communicate with OBS, for example, the client was unable to access the network.");  
    console.log(error);  
  };  
};  
createBucket();
```

5.2 列举桶列表(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

OBS桶是对象的容器，您上传的文件都存放在桶中。调用获取桶列表接口，可列举当前账号所有区域下符合指定条件的桶。返回结果以桶名的字典序排序。

接口约束

- 您必须拥有obs:bucket:ListAllMyBuckets权限，才能获取桶列表。建议使用IAM进行授权，配置方式详见[使用IAM自定义策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

ObsClient.listBuckets(params)

请求参数说明

表 5-8 请求参数列表

参数名称	参数类型	是否必选	描述
QueryLocation	boolean	可选	参数解释： 是否同时查询桶的区域位置。 约束限制： 无 取值范围： <ul style="list-style-type: none">• true：同时查询桶的区域位置• false：不查询桶的区域位置 默认取值： false

参数名称	参数类型	是否必选	描述
BucketType	string	可选	参数解释: 列举指定类型的桶。 约束限制: 无 取值范围: 桶类型取值可参见 BucketType 。 默认取值: 无, 如果不赋值则默认获取所有桶和并行文件系统列表。

表 5-9 BucketType

可选值	说明
OBJECT	对应类型为对象存储桶。
POSIX	对应类型为并行文件系统 (POSIX) 。

返回结果说明

表 5-10 返回结果

参数类型	描述
表5-11 说明 该接口返回是一个Promise类型, 需要使用Promise、async/await语法处理。	参数解释: 接口返回信息, 详见 表 5-11 。

表 5-11 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。

参数名称	参数类型	描述
InterfaceResult	ListBucketsOutput	<p>参数解释: 操作成功后的结果数据，详见 ListBucketsOutput。</p> <p>约束限制: 当Status大于300时为空。</p>

表 5-12 ICommonMsg

参数名称	参数类型	描述
Status	number	<p>参数解释: OBS服务端返回的HTTP状态码。</p> <p>取值范围: 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。</p>
Code	string	<p>参数解释: OBS服务端返回的错误码。</p>
Message	string	<p>参数解释: OBS服务端返回的错误描述。</p>
HostId	string	<p>参数解释: OBS服务端返回的请求服务端ID。</p>
RequestId	string	<p>参数解释: OBS服务端返回的请求ID。</p>
Id2	string	<p>参数解释: OBS服务端返回的请求ID2。</p>
Indicator	string	<p>参数解释: OBS服务端返回的详细错误码。</p>

表 5-13 ListBucketsOutput

参数名称	参数类型	描述
RequestId	string	<p>参数解释: OBS服务端返回的请求ID。</p> <p>默认取值: 无</p>

参数名称	参数类型	描述
Owner	Owner	参数解释: 桶的所有者。 取值范围: 桶所有者取值参见 Owner 。
Buckets	Bucket[]	参数解释: 桶信息列表。 取值范围: 桶信息取值参见 Bucket 。

表 5-14 Owner

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时必选	参数解释: 所有者的账号ID, 即domain_id。 取值范围: 如何获取账号ID请参见 如何获取账号ID和用户ID? (Node.js SDK) 默认取值: 无
Display Name	string	可选	参数解释: 所有者的账号用户名。 默认取值: 无

表 5-15 Bucket

参数名称	参数类型	描述
BucketName	string	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用类IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 <p>默认取值: 无</p>
CreationDate	string	<p>参数解释: 桶的创建时间。</p> <p>默认取值: 无</p>
Location	string	<p>参数解释: 桶所在的区域。</p> <p>约束限制: 该参数定义了桶将会被创建在哪个区域，如果使用的终端节点是obs.myhuaweicloud.com，可以不携带此参数；如果使用的终端节点不是obs.myhuaweicloud.com，则必须携带此参数。</p> <p>取值范围: 当前有效的OBS区域和终端节点的更多信息，请参考地区和终端节点。终端节点即调用API的请求地址，不同服务不同区域的终端节点不同，您可以向企业管理员获取区域和终端节点信息。</p> <p>默认取值: 终端节点为obs.myhuaweicloud.com且用户未设定区域时，默认为华北-北京一（cn-north-1）。</p>

参数名称	参数类型	描述
BucketType	string	参数解释: 桶类型。 取值范围: <ul style="list-style-type: none">对象桶: OBJECT并行文件系统: POSIX。 默认取值: 无

代码示例

可以通过ObsClient.listBuckets列举桶。以下代码展示如何列举所有桶列表。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listBuckets() {
  try {
    const params = {
      // 指定桶列表中是否存在Location字段, 此处以true为例, 默认是false
      QueryLocation: true,
      // 指定桶类型, 此处以为对象桶为例, 即获取对象桶列表, 默认不设置则获取所有对象桶和并行文件系统列表
      BucketType: "OBJECT",
    };
    // 列举桶列表
    const result = await obsClient.listBuckets(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List buckets successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log('Owner:');
      console.log('ID: %s', result.InterfaceResult.Owner.ID);
      console.log('Name: %s', result.InterfaceResult.Owner.Name);
      console.log('Buckets:');
      for (let i = 0; i < result.InterfaceResult.Buckets.length; i++) {
        const val = result.InterfaceResult.Buckets[i];
        console.log("Bucket[%d]-Name:%s,CreationDate:%s,Location: %s",
          i, val.BucketName, val.CreationDate, val.Location);
      }
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
  }
}
```

```
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
listBuckets();
```

5.3 判断桶是否存在(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

判断指定桶名的桶是否存在，返回的结果中HTTP状态码为200表明桶存在，否则返回404表明桶不存在。

接口约束

- 您必须是桶拥有者或拥有判断桶是否存在的权限，才能判断桶是否存在。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:HeadBucket权限，如果使用桶策略则需授予HeadBucket权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.headBucket(params)
```

请求参数

表 5-16 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

返回结果

表 5-17 返回结果

参数类型	描述
<p>表5-18</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表5-18。</p>

表 5-18 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	BaseResponseOutput	参数解释: 操作成功后的结果数据, 详见 BaseResponseOutput 。 约束限制: 当Status大于300时为空。

表 5-19 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-20 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

您可以通过ObsClient.headBucket接口判断该桶是否已存在。以下代码展示如何判断指定桶examplebucket是否存在。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function headBucket() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket"
    };
    // 判断桶是否存在
    const result = await obsClient.headBucket(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Head bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

headBucket();
```

5.4 删除桶(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

桶为空时，用户可以删除桶，以免占用桶数量配额。删除桶后需要等待30分钟才能创建同名桶。

说明

华为云无法恢复用户主动删除的OBS数据。因此调用接口删除桶后，桶相关数据无法恢复，请谨慎操作。

接口约束

- 待删除的桶必须为空，桶为空包含两方面含义：
 - 桶内没有任何对象，没有对象的任何历史版本，没有对象的删除标记（删除标记也视作一个历史版本）。
 - 桶内没有任何未合并的多段上传任务，即桶内不存在碎片。
- 删除桶非幂等操作，删除不存在的桶会失败。
- 您必须是桶拥有者或拥有删除桶的权限，才能删除桶。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:DeleteBucket权限，如果使用桶策略则需授予DeleteBucket权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.deleteBucket(params)
```

请求参数

表 5-21 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

返回结果

表 5-22 返回结果

参数类型	描述
<p>表5-23</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表5-23。</p>

表 5-23 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	BaseResponseOutput	参数解释: 操作成功后的结果数据，详见 BaseResponseOutput 。 约束限制: 当Status大于300时为空。

表 5-24 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-25 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

您可以通过ObsClient.deleteBucket删除桶。以下代码展示如何删除名为examplebucket的桶。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucket() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket"
    };
    // 删除桶
    const result = await obsClient.deleteBucket(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

deleteBucket();
```

5.5 获取桶元数据(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

调用获取桶元数据接口，可获取指定桶的相关信息，包括指定桶的存储类别、区域位置、跨域资源共享（CORS）规则、数据冗余存储策略等信息。

接口约束

- 您必须是桶拥有者或拥有获取桶元数据的权限，才能获取桶元数据。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:HeadBucket权限，如果使用桶策略则需授予HeadBucket权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getBucketMetadata(params)
```

请求参数

表 5-26 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

返回结果

表 5-27 返回结果

参数类型	描述
<p>表5-28</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表5-28。</p>

表 5-28 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	GetBucketMetadataOutput	参数解释: 操作成功后的结果数据, 详见 GetBucketMetadataOutput 。 约束限制: 当Status大于300时为空。

表 5-29 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-30 GetBucketMetadataOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。 默认取值: 无
StorageClass	StorageClassType	参数解释: 桶的存储类别。 取值范围: 详情参见 StorageClassType 。 默认取值: 当桶存储类别是标准存储时, 该值为空。
Location	string	参数解释: 桶所在的区域。 约束限制: 该参数定义了桶将会被创建在哪个区域, 如果使用的终端节点是obs.myhuaweicloud.com, 可以不携带此参数; 如果使用的终端节点不是obs.myhuaweicloud.com, 则必须携带此参数。 取值范围: 当前有效的OBS区域和终端节点的更多信息, 请参考 地区和终端节点 。终端节点即调用API的请求地址, 不同服务不同区域的终端节点不同, 您可以向企业管理员获取区域和终端节点信息。 默认取值: 终端节点为obs.myhuaweicloud.com且用户未设定区域时, 默认为华北-北京一 (cn-north-1)。
ObsVersion	string	参数解释: 桶所在的OBS服务版本号。 取值范围: <ul style="list-style-type: none">“3.0”表示最新版本的桶。“-”表示老版本的桶。 默认取值: 无

参数名称	参数类型	描述
AllowOrigin	string	<p>参数解释: 如果请求中的Origin满足桶的CORS规则, 则返回CORS规则中的AllowedOrigin。AllowedOrigin指定允许的跨域请求的来源, 即允许来自该域名下的请求访问该对象。</p> <p>约束限制: 表示域名的字符串, 每个匹配规则允许使用最多一个“*”通配符。例如: https://*.vbs.example.com。</p> <p>默认取值: 无</p>
AllowHeader	string	<p>参数解释: 如果请求的RequestHeader满足桶的CORS规则, 则返回CORS规则中的AllowedHeader。AllowedHeader是指允许的跨域请求的头域。只有匹配上允许的头域中的配置, 才被视为合法的CORS请求。</p> <p>约束限制: 最多可填写一个“*”通配符, 不支持&、:、<、空格以及中文字符。</p> <p>默认取值: 无</p>
AllowMethod	string	<p>参数解释: 桶CORS规则中的AllowedMethod。AllowedMethod指允许的跨域请求HTTP方法, 即桶和对象的几种操作类型。</p> <p>取值范围: 支持以下HTTP方法:</p> <ul style="list-style-type: none">• GET• PUT• HEAD• POST• DELETE <p>默认取值: 无</p>

参数名称	参数类型	描述
ExposeHeader	string	<p>参数解释: 桶CORS规则中的ExposeHeader。ExposeHeader是指CORS规则允许响应中可返回的附加头域，给客户端提供额外的信息。默认情况下浏览器只能访问以下头域：Content-Length、Content-Type，如果需要访问其他头域，需要在附加头域中配置。</p> <p>约束限制: 不支持*、&、:、<、空格以及中文字符。</p> <p>默认取值: 无</p>
MaxAgeSeconds	number	<p>参数解释: 桶CORS规则中的MaxAgeSeconds。MaxAgeSeconds指请求来源的客户端可以对跨域请求返回结果的缓存时间。</p> <p>约束限制: 每个CORS规则可以包含至多一个MaxAgeSeconds。</p> <p>取值范围: 0~(2³¹-1)，单位：秒。</p> <p>默认取值: 100，单位：秒。</p>
MultiEnterprise	string	<p>参数解释: 创桶时可指定的企业项目ID，开通企业项目的用户可以从企业项目服务获取。</p> <p>约束限制: Epid格式为uuid，未开通企业项目的用户可以不带该头域。</p> <p>示例：9892d768-2d13-450f-aac7-ed0e44c2585f</p> <p>取值范围: 如何获取企业项目ID参见如何获取企业项目ID。</p> <p>默认取值: 无</p>

参数名称	参数类型	描述
AzRedundancy	string	<p>可选</p> <p>参数解释: 创桶时可指定的数据冗余类型，即AZ类型。</p> <p>约束限制: 归档存储不支持多AZ。如果桶所在区域如果不支持多AZ存储，则该桶的存储类别默认为单AZ。</p> <p>取值范围: 如果将桶配置为多AZ，则取值为3az。如果将桶配置为单AZ，则无需选择此选项，系统默认赋值。</p> <p>默认取值: 单AZ</p>

表 5-31 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。</p>
ObsClient.enums.StorageClassWarm	WARM	<p>低频访问存储。</p> <p>低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。</p>
ObsClient.enums.StorageClassCold	COLD	<p>归档存储。</p> <p>归档存储适用于很少访问（平均一年访问一次）数据的业务场景。</p>

代码示例

您可以通过ObsClient.getBucketMetadata接口获取桶元数据。以下代码展示如何获取名为examplebucket桶的元数据信息。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
```

```
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露
// 风险。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
// security_token: process.env.SECURITY_TOKEN,
// endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketMetadata() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket"
    };
    // 获取桶元数据信息
    const result = await obsClient.getBucketMetadata(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s metadata successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageClass: %s", result.InterfaceResult.StorageClass);
      console.log("Location: %s", result.InterfaceResult.Location);
      console.log("AZRedundancy: %s", result.InterfaceResult.AZRedundancy);
      console.log("Epid: %s", result.InterfaceResult.Epid);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};
getBucketMetadata();
```

5.6 设置桶 ACL(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

OBS支持对桶操作进行权限控制，您可以为桶设置访问策略，指定某一个用户对某一个桶是否有权行使某一项指定操作。OBS权限控制的方式有IAM、桶策略和ACL三种，ACL按照粒度又分为桶ACL和对象ACL，本节将对桶ACL接口进行详细介绍，更多权限相关内容可参见《对象存储服务权限配置指南》的[OBS权限控制概述](#)章节。

桶ACL是跨账号场景的权限，设置授权的对象不是当前账号，也不是当前账号下的IAM用户，而是另一个华为云账号及其账号下的IAM用户；授权的范围是以桶为粒度的，一条ACL策略为一个桶设置策略，因此设置ACL策略时您必须明确指定桶名；桶ACL授予的权限包括桶的访问权限和桶ACL的访问权限两个方面，桶的访问权限包括对桶及桶内对象的查看和编辑权限，桶ACL的访问权限包括对桶ACL策略的查看和编辑权限，详情可参见[ACL权限控制方式介绍](#)。

调用设置桶ACL接口，您可以修改指定桶的ACL策略。

接口约束

- 单个桶最多支持100条ACL策略。
- 您必须是桶拥有者或拥有设置桶ACL的权限，才能设置桶ACL。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutBucketAcl权限，如果使用桶策略则需授予PutBucketAcl权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.setBucketAcl(params)
```

请求参数

表 5-32 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。• 桶命名规则如下：<ul style="list-style-type: none">- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。- 禁止使用IP地址。- 禁止以“-”或“.”开头及结尾。- 禁止两个“.”相邻（如：“my..bucket”）。- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。• 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
ACL	AclType	可选	<p>参数解释: 预定义访问策略。</p> <p>约束限制: Owner、Grants与ACL不能全为空。</p> <p>取值范围: AclType取值详见AclType。</p> <p>默认取值: 无</p>
Owner	Owner	可选	<p>参数解释: 桶的所有者。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • Owner和Grants必须配套使用，且与ACL互斥。 • Owner、Grants与ACL不能全为空。 <p>取值范围: 详见Owner。</p> <p>默认取值: 无</p>
Grants	Grant []	可选	<p>参数解释: 被授权用户权限信息。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • Owner和Grants必须配套使用，且与ACL互斥。 • Owner、Grants与ACL不能全为空。 <p>取值范围: 详见Grant。</p> <p>默认取值: 无</p>

表 5-33 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限，其他任何人都没有访问权限。

常量名	原始值	说明
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	桶公共读，桶内对象公共读。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadDelivered不能应用于对象。
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	桶公共读写，桶内对象公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadWriteDelivered不能应用于对象。

常量名	原始值	说明
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	<p>设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。</p> <p>默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。</p>

表 5-34 Owner

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时必选	<p>参数解释： 所有者的账号ID，即domain_id。</p> <p>取值范围： 如何获取账号ID请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值： 无</p>
Display Name	string	可选	<p>参数解释： 所有者的账号用户名。</p> <p>默认取值： 无</p>

表 5-35 Grant

参数名称	参数类型	是否必选	描述
Grantee	Grantee	作为请求参数时必选	<p>参数解释： 被授权用户相关信息，详见Grantee。</p>

参数名称	参数类型	是否必选	描述
Permission	PermissionType	作为请求参数时必选	<p>参数解释: 被授予的权限。</p> <p>取值范围: 权限取值范围详见PermissionType。</p> <p>默认取值: 无</p>
Delivered	boolean	可选	<p>参数解释: 桶内对象ACL是否继承桶的ACL。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • true: 桶内对象ACL继承桶的ACL。 • false: 桶内对象ACL不继承桶的ACL。 <p>默认取值: 无</p>

表 5-36 Grantee

参数名称	参数类型	是否必选	描述
Type	string	作为请求参数时必选	<p>参数解释: 被授权用户的类型。</p> <p>取值范围: 被授权用户类型的取值范围详见GranteeType。</p> <p>默认取值: 无</p>
ID	string	作为请求参数时, 如果Type为用户类型则必选。	<p>参数解释: 被授权用户的账号ID, 即domain_id。</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Name	string	作为请求参数时可选	<p>参数解释: 被授权用户的账号名。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 不支持中文。 只能以字母开头。 长度为6-32个字符。 只能包含英文字母、数字或特殊字符(-_)。 <p>默认取值: 无</p>
URI	Group UriType	作为请求参数时, 如果Type为用户组类型则必选。	<p>参数解释: 被授权的用户组。</p> <p>取值范围: 授权用户组取值范围详见GroupUriType。</p> <p>默认取值: 无</p>

表 5-37 GranteeType

常量值	说明
Group	授权给用户组。
CanonicalUser	授权给单个用户。

表 5-38 GroupUriType

常量名	原始值	说明
ObsClient.enums.GroupAllUsers	AllUsers	所有用户。
ObsClient.enums.GroupAuthenticatedUsers	AuthenticatedUsers	授权用户, 已废弃。
ObsClient.enums.GroupLogDelivery	LogDelivery	日志投递组, 已废弃。

表 5-39 PermissionType

常量名	原始值	说明
ObsClient.enums.PermissionRead	READ	如果有桶的读权限，则可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 如果有对象的读权限，则可以获取该对象内容和元数据。
ObsClient.enums.PermissionWrite	WRITE	如果有桶的写权限，则可以上传、覆盖和删除该桶内任何对象和段。 此权限在对象上不适用。
ObsClient.enums.PermissionReadAcp	READ_ACP	如果有读ACP的权限，则可以获取对应的桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有读对应桶或对象ACP的权限。
ObsClient.enums.PermissionWriteAcp	WRITE_ACP	如果有写ACP的权限，则可以更新对应桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。 拥有了写ACP的权限，由于可以更改权限控制策略，实际上意味着拥有了完全访问的权限。
ObsClient.enums.PermissionFullControl	FULL_CONTROL	如果有桶的完全控制权限意味着拥有PermissionRead、PermissionWrite、PermissionReadAcp和PermissionWriteAcp的权限。 如果有对象的完全控制权限意味着拥有PermissionRead、PermissionReadAcp和PermissionWriteAcp的权限。

返回结果

表 5-40 返回结果

参数类型	描述
<p>表5-41</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表5-41。</p>

表 5-41 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	表12 BaseResponse Output	参数解释: 操作成功后的结果数据, 详见 表5-43 。 约束限制: 当Status大于300时为空。

表 5-42 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-43 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例：创桶时指定预定义访问策略

以下代码展示如何在创建桶时指定私有读写预定义访问策略。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function createBucket() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定存储桶所在区域，此处以“ap-southeast-1”为例，必须跟传入的Endpoint中Region保持一致
      // Location: "ap-southeast-1",
      // 指定存储桶的权限控制策略，此处以obs.AclPrivate为例
      ACL: obsClient.enums.AclPrivate,
    };
    // 创建桶
    const result = await obsClient.createBucket(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Create bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

createBucket();
```

代码示例：为桶设置预定义访问策略

本示例用于设置名为examplebucket桶的ACL权限控制为私有权限。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketAcl() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定桶的Acl, 此处以私有权限为例
      ACL: obsClient.enums.AclPrivate
    };
    // 设置桶ACL
    const result = await obsClient.setBucketAcl(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s acl successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }

    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

setBucketAcl();
```

代码示例：直接设置桶访问权限

本示例用于设置名为examplebucket桶的ACL权限控制为, 所有用户拥有公共读的权限, 指定用户(0a03f5833900d3730f13c00f49d5exxx)拥有公共写的权限。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
```

```
// 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
// security_token: process.env.SECURITY_TOKEN,
// endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketAcl() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定存储桶所有者ID，此处以ownerid为例
      Owner: { ID: 'ownerid' },
      Grants: [
        // 指定所有用户拥有公共读的权限
        { Grantee: { Type: 'Group', URL: obsClient.enums.GroupAllUsers }, Permission:
obsClient.enums.PermissionRead },
        // 指定特定用户拥有公共写的权限，此处用户ID为0a03f5833900d3730f13c00f49d5exxx
        { Grantee: { Type: 'CanonicalUser', ID: '0a03f5833900d3730f13c00f49d5exxx' }, Permission:
obsClient.enums.PermissionWrite },
      ]
    };
    // 设置桶ACL
    const result = await obsClient.setBucketAcl(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s acl successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

setBucketAcl();
```

5.7 获取桶 ACL(Node.js SDK)

功能介绍

OBS支持对桶操作进行权限控制，您可以为桶设置访问策略，指定某一个用户对某一个桶是否有权行使某一项指定操作。OBS权限控制的方式有IAM、桶策略和ACL三种，ACL按照粒度又分为桶ACL和对象ACL，本节将对桶ACL接口进行详细介绍，更多权限相关内容可参见《对象存储服务权限配置指南》的[OBS权限控制概述](#)章节。

桶ACL是跨账号场景的权限，设置授权的对象不是当前账号，也不是当前账号下的IAM用户，而是另一个账号及其账号下的IAM用户；授权的范围是以桶为粒度的，一条ACL策略为一个桶设置策略，因此设置ACL策略时您必须明确指定桶名；桶ACL授予的权限包括桶的访问权限和桶ACL的访问权限两个方面，桶的访问权限包括对桶及桶内对象的

查看和编辑权限，桶ACL的访问权限包括对桶ACL策略的查看和编辑权限，详情可参见[ACL权限控制方式介绍](#)。

调用获取桶ACL接口，您可以获取指定桶的ACL策略。

接口约束

- 您必须是桶所有者或拥有获取桶ACL的权限，才能获取桶ACL。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:GetBucketAcl权限，如果使用桶策略则需授予GetBucketAcl权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getBucketAcl(params)
```

请求参数

表 5-44 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 5-45 返回结果

参数类型	描述
表5-46 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见表5-46。

表 5-46 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表5-48	参数解释： 操作成功后的结果数据，详见 表5-48 。 约束限制： 当Status大于300时为空。

表 5-47 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释： OBS服务端返回的HTTP状态码。 取值范围： 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 。
Code	string	参数解释： OBS服务端返回的错误码。
Message	string	参数解释： OBS服务端返回的错误描述。
HostId	string	参数解释： OBS服务端返回的请求服务端ID。
RequestId	string	参数解释： OBS服务端返回的请求ID。

Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-48 GetBucketAclOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
Owner	Owner	参数解释: 桶的所有者的账号ID, 详见 Owner 。
Grants	Grant[]	参数解释: 被授权用户权限信息, 详见 Grant 。

表 5-49 Owner

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时必选	参数解释: 所有者的账号ID, 即domain_id。 取值范围: 如何获取账号ID请参见 如何获取账号ID和用户ID? (Node.js SDK) 默认取值: 无
Display Name	string	可选	参数解释: 所有者的账号用户名。 默认取值: 无

表 5-50 Grant

参数名称	参数类型	描述
Grantee	Grantee	参数解释: 被授权用户相关信息, 详见 Grantee 。

参数名称	参数类型	描述
Permission	string	参数解释： 被授予的权限，详见 PermissionType 。

表 5-51 Grantee

参数名称	参数类型	描述
Type	string	参数解释： 被授权用户的类型，详见 GranteeType 。
ID	string	参数解释： 被授权用户的账号ID，即domain_id。
Name	string	参数解释： 被授权用户的账号名。
URI	string	参数解释： 被授权的用户组，详见 GroupUriType 。

表 5-52 GranteeType

常量值	说明
Group	授权给用户组。
CanonicalUser	授权给单个用户。

表 5-53 GroupUriType

常量名	原始值	说明
ObsClient.enums.GroupAllUsers	AllUsers	所有用户。
ObsClient.enums.GroupAuthenticatedUsers	AuthenticatedUsers	授权用户，已废弃。
ObsClient.enums.GroupLogDelivery	LogDelivery	日志投递组，已废弃。

表 5-54 PermissionType

常量名	原始值	说明
ObsClient.enums.PermissionRead	READ	如果有桶的读权限，则可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 如果有对象的读权限，则可以获取该对象内容和元数据。
ObsClient.enums.PermissionWrite	WRITE	如果有桶的写权限，则可以上传、覆盖和删除该桶内任何对象和段。 此权限在对象上不适用。
ObsClient.enums.PermissionReadAcp	READ_ACP	如果有读ACP的权限，则可以获取对应的桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有读对应桶或对象ACP的权限。
ObsClient.enums.PermissionWriteAcp	WRITE_ACP	如果有写ACP的权限，则可以更新对应桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。 拥有了写ACP的权限，由于可以更改权限控制策略，实际上意味着拥有了完全访问的权限。
ObsClient.enums.PermissionFullControl	FULL_CONTROL	如果有桶的完全控制权限意味着拥有PermissionRead、PermissionWrite、PermissionReadAcp和PermissionWriteAcp的权限。 如果有对象的完全控制权限意味着拥有PermissionRead、PermissionReadAcp和PermissionWriteAcp的权限。

代码示例

本示例用于获取名为examplebucket桶的ACL权限控制列表信息。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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
  // 您可以通过登录访问管理控制台获取访问密钥AK/SK，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketAcl() {
```

```
try {
  const params = {
    // 指定存储桶名称
    Bucket: "examplebucket"
  };
  // 获取桶ACL
  const result = await obsClient.getBucketAcl(params);
  if (result.CommonMsg.Status <= 300) {
    console.log("Get bucket(%s)'s acl rules successful!", params.Bucket);
    console.log("RequestId: %s", result.InterfaceResult.RequestId);
    console.log("Owner[ID]: %s", result.InterfaceResult.Owner.ID);
    console.log("Grants:");
    for (let i = 0; i < result.InterfaceResult.Grants.length; i++) {
      const grant = result.InterfaceResult.Grants[i];
      console.log("Grant[%d]-Type:%s, ID:%s, URI:%s, Permission:%s",
        i, grant.Grantee.Type, grant.Grantee.ID, grant.Grantee.URI, grant.Permission
      );
    };
    return;
  };
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
  console.log("Status: %d", result.CommonMsg.Status);
  console.log("Code: %s", result.CommonMsg.Code);
  console.log("Message: %s", result.CommonMsg.Message);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};

getBucketAcl();
```

相关链接

- 关于获取桶ACL的API说明，请参见[获取桶ACL](#)。
- 获取桶ACL过程中返回的错误码含义、问题原因及处理措施可参考[OBS错误码](#)。
- 权限相关常见问题请参见[权限相关常见问题](#)。

5.8 设置桶策略(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

OBS支持对桶操作进行权限控制，您可以为桶设置访问策略，指定某一个用户对某一个桶是否有权行使某一项指定操作。OBS权限控制的方式有IAM、桶策略和ACL三种，本节将对桶策略接口进行详细介绍，更多权限相关内容可参见《对象存储服务权限配置指南》的[OBS权限控制概述](#)章节。

桶策略是作用于所配置的OBS桶及桶内对象的，您可以通过桶策略可为IAM用户或其他账号授权桶及桶内对象的操作权限。允许其他账号访问OBS资源，可以使用桶策略的方式授权对应权限。当不同的桶对于不同的IAM用户有不同的访问控制需求时，需使用桶策略分别授权IAM用户不同的权限。

调用设置桶策略接口，您可为指定桶设置桶策略。

接口约束

- 创建桶和获取桶列表这两个服务级的操作权限，需要通过[IAM权限](#)配置。
- 由于缓存的存在，配置桶策略后，最长需要等待5分钟策略才能生效。
- 您必须是桶拥有者或拥有设置桶策略的权限，才能设置桶策略。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutBucketPolicy权限，如果使用桶策略则需授予PutBucketPolicy权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.setBucketPolicy(params)
```

请求参数

表 5-55 请求参数列表

参数名称	类型	是否必选	说明
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。• 桶命名规则如下：<ul style="list-style-type: none">- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。- 禁止使用IP地址。- 禁止以“-”或“.”开头及结尾。- 禁止两个“.”相邻（如：“my..bucket”）。- 禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。• 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	类型	是否必选	说明
Policy	string	必选	<p>参数解释: 策略信息, JSON格式的字符串。</p> <p>约束限制:</p> <ul style="list-style-type: none"> Policy中的Resource字段包含的桶名必须和当前设置桶策略的桶名一致。 Policy具体格式请参考Policy格式 <p>取值范围: 请参考Policy格式</p> <p>默认取值: 无</p>

返回结果

表 5-56 返回结果

参数类型	描述
<p>表5-57</p> <p>说明 该接口返回是一个Promise类型, 需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息, 详见表5-57。</p>

表 5-57 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	<p>参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见ICommonMsg。</p>
InterfaceResult	表5-59	<p>参数解释: 操作成功后的结果数据, 详见表5-59。</p> <p>约束限制: 当Status大于300时为空。</p>

表 5-58 ICommonMsg

参数名称	参数类型	描述
------	------	----

Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-59 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例：设置桶策略

您可以通过`ObsClient.setBucketPolicy`设置桶策略。代码示例如下：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
```

```
// endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketPolicy() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定桶策略
      Policy: "{\"Statement\":[{\"Sid\":\"自定义策略-2482\",\"Effect\":\"Allow\",\"Principal\":{\"ID\":\"*\"}],\"Action\":[\"*\"],\"ListBucket\"],\"Resource\":[\"examplebucket\"]}",
    };
    // 设置桶策略
    const result = await obsClient.setBucketPolicy(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s policy successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

setBucketPolicy();
```

5.9 获取桶策略(Node.js SDK)

功能介绍

OBS支持对桶操作进行权限控制, 您可以为桶设置访问策略, 指定某一个用户对某一个桶是否有权行使某一项指定操作。OBS权限控制的方式有IAM、桶策略和ACL三种, 本节将对桶策略接口进行详细介绍, 更多权限相关内容可参见《对象存储服务权限配置指南》的[OBS权限控制概述](#)章节。

桶策略是作用于所配置的OBS桶及桶内对象的, 您可以通过桶策略可为IAM用户或其他账号授权桶及桶内对象的操作权限。允许其他账号访问OBS资源, 可以使用桶策略的方式授权对应权限。当不同的桶对于不同的IAM用户有不同的访问控制需求时, 需使用桶策略分别授权IAM用户不同的权限。

调用获取桶策略接口, 您可获取指定桶的桶策略。

接口约束

- 以下两种场景无法使用此接口获取桶策略, 系统将返回“404 NoSuchBucketPolicy”的错误:
 - 指定桶的策略不存在
 - 指定桶的标准桶策略为私有且未设置高级桶策略
- 您必须是桶拥有者或拥有获取桶策略的权限, 才能获取桶策略。建议使用IAM或桶策略进行授权, 如果使用IAM则需授予obs:bucket:GetBucketPolicy权限, 如果

使用桶策略则需授予GetBucketPolicy权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。

- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

ObsClient.getBucketPolicy(params)

请求参数

表 5-60 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。• 桶命名规则如下：<ul style="list-style-type: none">- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。- 禁止使用IP地址。- 禁止以“-”或“.”开头及结尾。- 禁止两个“.”相邻（如：“my..bucket”）。- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。• 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 5-61 返回结果

参数类型	描述
表5-62 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见表5-62。

表 5-62 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表5-64	参数解释： 操作成功后的结果数据，详见 表5-64 。 约束限制： 当Status大于300时为空。

表 5-63 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释： OBS服务端返回的HTTP状态码。 取值范围： 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 。
Code	string	参数解释： OBS服务端返回的错误码。
Message	string	参数解释： OBS服务端返回的错误描述。
HostId	string	参数解释： OBS服务端返回的请求服务端ID。
RequestId	string	参数解释： OBS服务端返回的请求ID。

Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-64 GetBucketPolicyOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
Policy	string	参数解释: 策略信息，JSON格式的字符串。 约束限制: <ul style="list-style-type: none"> Policy中的Resource字段包含的桶名必须和当前设置桶策略的桶名一致。 Policy具体格式请参考Policy格式。

代码示例

本示例用于获取名为examplebucket的桶策略。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketPolicy() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket"
    };
    // 获取桶策略
    const result = await obsClient.getBucketPolicy(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s policy successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("Policy: %s", result.InterfaceResult.Policy);
    }
  }
}
```

```
return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
console.log(error);
};
};
};
getBucketPolicy();
```

5.10 删除桶策略(Node.js SDK)

功能介绍

OBS支持对桶操作进行权限控制，您可以为桶设置访问策略，指定某一个用户对某一个桶是否有权行使某一项指定操作。OBS权限控制的方式有IAM、桶策略和ACL三种，本节将对桶策略接口进行详细介绍，更多权限相关内容可参见《对象存储服务权限配置指南》的[OBS权限控制概述](#)章节。

桶策略是作用于所配置的OBS桶及桶内对象的，您可以通过桶策略可为IAM用户或其他账号授权桶及桶内对象的操作权限。允许其他账号访问OBS资源，可以使用桶策略的方式授权对应权限。当不同的桶对于不同的IAM用户有不同的访问控制需求时，需使用桶策略分别授权IAM用户不同的权限。

调用删除桶策略，您可删除指定桶的桶策略。无论桶的策略本身是否存在，删除成功后系统都直接返回“204 No Content”的结果。

接口约束

- 您必须是桶拥有者或拥有删除桶策略的权限，才能删除桶策略。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:DeleteBucketPolicy权限，如果使用桶策略则需授予DeleteBucketPolicy权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.deleteBucketPolicy(params)
```

请求参数

表 5-65 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 5-66 返回结果

参数类型	描述
<p>表5-67</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表5-67。</p>

表 5-67 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	表5-69	参数解释: 操作成功后的结果数据, 详见 表5-69 。 约束限制: 当Status大于300时为空。

表 5-68 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-69 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释： OBS服务端返回的请求ID。

代码示例

本示例用于删除名为examplebucket的桶策略。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucketPolicy() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket"
    };
    // 删除桶策略
    const result = await obsClient.deleteBucketPolicy(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s policy successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("Delete bucket(%s)'s policy fail!", params.Bucket);
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

deleteBucketPolicy();
```

相关链接

- 关于删除桶策略的API说明，请参见[删除桶策略](#)。

- 删除桶策略过程中返回的错误码含义、问题原因及处理措施可参考[OBS错误码](#)。
- 权限相关常见问题请参见[权限相关常见问题](#)。

5.11 获取桶区域位置(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

获取指定桶所在的区域位置。

接口约束

- 您必须是桶拥有者或拥有获取桶区域位置的权限，才能获取桶区域位置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:GetBucketLocation权限，如果使用桶策略则需授予GetBucketLocation权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getBucketLocation(params)
```

请求参数

表 5-70 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 5-71 返回结果

参数类型	描述
<p>表5-72</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表5-72。</p>

表 5-72 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	GetBucketMetadataOutput	参数解释: 操作成功后的结果数据, 详见 GetBucketMetadataOutput 。 约束限制: 当Status大于300时为空。

表 5-73 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-74 GetBucketLocationOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
Location	string	参数解释: 桶所在的区域。 约束限制: 该参数定义了桶将会被创建在哪个区域，如果使用的终端节点是obs.myhuaweicloud.com，可以不携带此参数；如果使用的终端节点不是obs.myhuaweicloud.com，则必须携带此参数。 取值范围: 当前有效的OBS区域和终端节点的更多信息，请参考 地区和终端节点 。终端节点即调用API的请求地址，不同服务不同区域的终端节点不同，您可以向企业管理员获取区域和终端节点信息。 默认取值: 终端节点为obs.myhuaweicloud.com且用户未设定区域时，默认为华北-北京一（cn-north-1）。

代码示例

您可以通过ObsClient.getBucketLocation获取桶的区域位置。以下代码展示如何获取名为examplebucket桶区域位置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketLocation() {
  try {
    const params = {
      // 指定存储桶名称
```

```
Bucket: "examplebucket"
};
// 获取桶区域位置
const result = await obsClient.getBucketLocation(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Get bucket(%s)'s location successful!", params.Bucket);
  console.log("Location: %s", result.InterfaceResult.Location);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};

getBucketLocation();
```

📖 说明

创建桶时可以指定桶的区域位置，请参见[创建桶](#)。

5.12 获取桶存量信息(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

调用获取桶存量信息接口，可查询指定桶内的对象个数，以及对象占用空间的大小。

📖 说明

由于OBS桶存量是后台统计，因此存量会有一些的时延，不能实时更新，因此不建议对存量做实时校验。

接口约束

- 您必须是桶拥有者或拥有获取桶存量信息的权限，才能获取桶存量信息。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:GetBucketStorage权限，如果使用桶策略则需授予GetBucketStorage权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getBucketStorageInfo(params)
```

请求参数

表 5-75 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 5-76 返回结果

参数类型	描述
<p>表5-77</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表5-77。</p>

表 5-77 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表5-79	参数解释: 操作成功后的结果数据，详见 表5-79 。 约束限制: 当Status大于300时为空。

表 5-78 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-79 GetBucketStorageInfoOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
Size	number	参数解释: 桶中对象占用的空间大小。
ObjectNumber	number	参数解释: 桶内对象个数。

代码示例

桶存量信息包括桶的空间大小以及桶包含的对象个数。您可以通过 `ObsClient.getBucketStorageInfo` 获取桶的存量信息。以下代码展示如何获取名为 `examplebucket` 桶的存量信息。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketStorageInfo() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket"
    };
    // 获取桶存量信息
    const result = await obsClient.getBucketStorageInfo(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s storage-info successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("Size: %s", result.InterfaceResult.Size);
      console.log("ObjectNumber: %s", result.InterfaceResult.ObjectNumber);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
```

```
console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
console.log(error);
};
};
getBucketStorageInfo();
```

5.13 设置桶配额(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

桶配额是桶容量的上限值。默认情况下，OBS系统和单个桶都没有总数据容量和对象数量的限制。您可以根据需要，为桶设置配额限制来控制桶内允许上传的对象总容量，超过设置的对象容量后，上传对象会失败。例如您为桶设置配额为100G，那么当桶内所有对象的大小总和达到100G后，再上传对象就会因为超过配额导致上传失败。

容量限制只对桶配额设置生效后的对象上传操作有影响，如果您设置桶配额时，桶配额的数值小于桶中已上传的对象容量，并不会导致桶中已有对象被删除，但该桶后续将不能再上传任何对象。这种情况下只有删除部分已有对象，将已用空间释放到配额限制以下之后才能再次上传新对象。

接口约束

- 桶配额值必须为非负整数，单位为字节，支持的最大值为 $2^{63}-1$ 。
- OBS没有提供删除桶配额的接口，您可以将桶配额设置为0来取消配额限制。
- 您必须是桶拥有者或拥有设置桶配额的权限，才能设置桶配额。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutBucketQuota权限，如果使用桶策略则需授予PutBucketQuota权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.setBucketQuota(params)
```

请求参数

表 5-80 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>
StorageQuota	number	必选	<p>参数解释: 桶的配额值。</p> <p>约束限制: 无</p> <p>取值范围: 0~(2⁶³-1)，单位：字节。</p> <p>默认取值: 0，表示没有限制桶配额。</p>

返回结果

表 5-81 返回结果

参数类型	描述
表5-82 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见表5-82。

表 5-82 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表5-84	参数解释： 操作成功后的结果数据，详见 表5-84 。 约束限制： 当Status大于300时为空。

表 5-83 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释： OBS服务端返回的HTTP状态码。 取值范围： 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 。
Code	string	参数解释： OBS服务端返回的错误码。
Message	string	参数解释： OBS服务端返回的错误描述。
HostId	string	参数解释： OBS服务端返回的请求服务端ID。
RequestId	string	参数解释： OBS服务端返回的请求ID。

Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-84 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

您可以通过ObsClient.setBucketQuota设置桶配额。以下代码展示如何设置名为examplebucket的桶配额为1G。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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
  // 您可以通过登录访问管理控制台获取访问密钥AK/SK, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketQuota() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定指定桶的配额值, 此处以1G为例, 单位为字节
      StorageQuota: 1024 * 1024 * 1024
    };
    // 设置桶配额
    const result = await obsClient.setBucketQuota(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s quota successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  }
}
```

```
} catch (error) {  
  console.log("An Exception was found, which means the client encountered an internal problem when  
  attempting to communicate with OBS, for example, the client was unable to access the network.");  
  console.log(error);  
};  
};  
setBucketQuota();
```

📖 说明

- 使用**StorageQuota**参数指定桶的配额大小。
- 桶配额值必须为非负整数，单位为字节，支持的最大值为 $2^{63} - 1$ 。

5.14 获取桶配额(Node.js SDK)

功能说明

桶配额是桶容量的上限值。调用获取桶配额接口，可获取指定桶的配额值。桶配额为0代表桶容量没有上限。

接口约束

- 桶配额值必须为非负整数，单位为字节，支持的最大值为 $2^{63}-1$ 。
- 桶的拥有者的状态是欠费冻结状态时不可以查询桶配额信息。
- 您必须是桶拥有者或拥有获取桶配额的权限，才能获取桶配额。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:GetBucketQuota权限，如果使用桶策略则需授予GetBucketQuota权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getBucketQuota(params)
```

请求参数

表 5-85 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

返回结果

表 5-86 返回结果

参数类型	描述
<p>表5-87</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表5-87。</p>

表 5-87 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表5-89	参数解释: 操作成功后的结果数据，详见 表5-89 。 约束限制: 当Status大于300时为空。

表 5-88 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-89 GetBucketQuotaOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
StorageQuota	number	参数解释: 桶的配额值。 默认取值: 0, 表示没有限制桶配额。

代码示例

您可以通过ObsClient.getBucketQuota获取桶配额。以下代码展示如何获取名为examplebucket的桶配额。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AK/SK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketQuota() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket"
    };
    // 获取桶配额
    const result = await obsClient.getBucketQuota(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s quota successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageQuota: %s", result.InterfaceResult.StorageQuota);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}
```

```
};  
getBucketQuota();
```

5.15 设置桶存储类别(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

调用设置桶存储类别接口，可设置指定桶的存储类别。设置了桶的默认存储类别之后，若上传对象、复制对象和初始化多段上传任务时未指定对象的存储类别，则该对象的存储类别默认与桶的存储类别保持一致。

接口约束

- 您必须是桶拥有者或拥有设置桶存储类别的权限，才能设置桶存储类别。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:PutBucketStoragePolicy权限，如果使用桶策略则需授予PutBucketStoragePolicy权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.setBucketStoragePolicy(params)
```

请求参数

表 5-90 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>
StorageClass	StorageClassType	必选	<p>参数解释: 桶的存储类别。</p> <p>约束限制: 无</p> <p>取值范围: 存储类别取值详见StorageClassType。</p> <p>默认取值: 无</p>

表 5-91 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

返回结果

表 5-92 返回结果

参数类型	描述
<p>表5-93</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表5-93。</p>

表 5-93 Response

参数名称	参数类型	描述
CommonMsg	ICCommonMsg	<p>参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICCommonMsg。</p>
InterfaceResult	表5-95	<p>参数解释： 操作成功后的结果数据，详见表5-95。</p> <p>约束限制： 当Status大于300时为空。</p>

表 5-94 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-95 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

您可以通过ObsClient.setBucketStoragePolicy设置桶存储类别。以下代码展示如何设置名为examplebucket桶的存储类别为低频访问存储。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
});
```

```
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露
// 风险。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
// security_token: process.env.SECURITY_TOKEN,
// endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketStoragePolicy() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定桶的存储类别，此处以obsClient.enums.StorageClassWarm为例
      StorageClass: obsClient.enums.StorageClassWarm
    };
    // 设置桶存储类别
    const result = await obsClient.setBucketStoragePolicy(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s storage-class successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

setBucketStoragePolicy();
```

5.16 获取桶存储类别(Node.js SDK)

功能介绍

调用获取桶存储类别接口，可获取指定桶的存储类别。

接口约束

- 您必须是桶拥有者或拥有获取桶存储类别的权限，才能获取桶存储类别。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:GetBucketStoragePolicy权限，如果使用桶策略则需授予GetBucketStoragePolicy权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getBucketStoragePolicy(params)
```

请求参数

表 5-96 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 5-97 返回结果

参数类型	描述
<p>表5-98</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表5-98。</p>

表 5-98 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	GetBucketMetadataOutput	参数解释: 操作成功后的结果数据, 详见 GetBucketMetadataOutput 。 约束限制: 当Status大于300时为空。

表 5-99 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 5-100 GetBucketStorageClassOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
StorageClass	StorageClass Type	参数解释: 桶的存储类别。 取值范围: 存储类别取值详见 StorageClassType 。

表 5-101 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

代码示例

您可以通过ObsClient.getBucketStoragePolicy获取桶存储类别。以下代码展示如何获取名为examplebucket桶的存储类别。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AK/SK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
```

```
});  
  
async function getBucketStoragePolicy() {  
  try {  
    const params = {  
      // 指定存储桶名称  
      Bucket: "examplebucket"  
    };  
    // 获取桶的存储类别  
    const result = await obsClient.getBucketStoragePolicy(params);  
    if (result.CommonMsg.Status <= 300) {  
      console.log("Get bucket(%s)'s storage-class successful!", params.Bucket);  
      console.log("RequestId: %s", result.CommonMsg.RequestId);  
      console.log('StorageClass: %s', result.InterfaceResult.StorageClass);  
      return;  
    };  
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");  
    console.log("Status: %d", result.CommonMsg.Status);  
    console.log("Code: %s", result.CommonMsg.Code);  
    console.log("Message: %s", result.CommonMsg.Message);  
    console.log("RequestId: %s", result.CommonMsg.RequestId);  
  } catch (error) {  
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");  
    console.log(error);  
  };  
};  
  
getBucketStoragePolicy();
```

相关链接

- 关于获取桶存储类别的API说明，请参见[获取桶存储类型](#)。
- 获取桶存储类别过程中返回的错误码含义、问题原因及处理措施可参考[OBS错误码](#)。
- 桶和对象相关常见问题请参见[桶和对象相关常见问题](#)。

6 上传对象(Node.js SDK)

6.1 对象上传简介(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

在OBS中，用户操作的基本数据单元是对象。OBS Node.js SDK提供了丰富的对象上传接口，可以通过以下方式上传对象：

- [文本上传\(Node.js SDK\)](#)
- [流式上传\(Node.js SDK\)](#)
- [文件上传\(Node.js SDK\)](#)
- [初始化多段上传任务\(Node.js SDK\)](#)
- [追加上传\(Node.js SDK\)](#)
- [断点续传上传\(Node.js SDK\)](#)
- [基于表单上传\(Node.js SDK\)](#)

SDK支持上传0KB~5GB的对象。文本上传、流式上传、文件上传和追加上传每次上传内容大小不能超过5GB；当上传较大文件时，请使用分段上传，分段上传每段内容大小不能超过5GB；基于表单上传提供了基于浏览器表单上传对象的方式。

如果上传的对象权限设置为匿名用户读取权限，对象上传成功后，匿名用户可通过链接地址访问该对象数据。对象链接地址格式为：<https://桶名.域名/文件夹目录层级/对象名>。如果该对象存在于桶的根目录下，则链接地址将不需要有文件夹目录层级。

6.2 文本上传(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

本章节介绍上传单个对象到指定桶中。

接口约束

- 您必须是桶拥有者或拥有上传对象的权限，才能上传对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObject权限，如果使用桶策略则需授予PutObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 单次上传对象大小范围是[0, 5GB]。
- 如果需要上传超过5GB的大文件，需要使用[分段上传](#)。

方法定义

```
ObsClient.putObject(params)
```

请求参数

表 6-1 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。• 桶命名规则如下：<ul style="list-style-type: none">- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。- 禁止使用IP地址。- 禁止以“-”或“.”开头及结尾。- 禁止两个“.”相邻（如：“my.bucket”）。- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。• 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
ACL	AclType	可选	<p>参数解释: 创建对象时可指定的预定义访问策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见ACL功能介绍。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的访问策略选项参见AclType。</p> <p>默认取值: 无</p>
StorageClass	StorageClassType	可选	<p>参数解释: 创建对象时，可以加上此头域设置对象的存储类别。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的存储类别参见StorageClassType。</p> <p>默认取值: 如果未设置此头域，则对象的存储列表默认为桶存储类别。</p>

参数名称	参数类型	是否必选	描述
Body	string stream.Readable	可选	<p>参数解释: 待上传段的内容，支持字符串和stream.Readable对象。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 单次上传对象大小范围是[0, 5GB]。 • 如果需要上传超过5GB的大文件，需要使用分段上传。 • Body与SourceFile不能同时使用。 <p>取值范围: 无</p> <p>默认取值: 无</p>
SourceFile	string	可选	<p>参数解释: 待上传对象的源文件路径。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 单次上传大小范围是[0, 5GB]。 • Body与SourceFile不能同时使用。 • Offset和SourceFile配套使用，用于指定上传源文件中的某一部分数据。 <p>取值范围: 无</p> <p>默认取值: 无</p>
Offset	number	可选	<p>参数解释: 源文件中某一分段的起始偏移大小。</p> <p>约束限制: Offset和SourceFile配套使用，用于指定上传源文件中的某一部分数据。</p> <p>取值范围: 非负整数，小于待上传对象的大小，单位：字节。</p> <p>默认取值: 0</p>

参数名称	参数类型	是否必选	描述
ProgressCallback	function	可选	<p>参数解释: 获取上传进度的回调函数。</p> <p>约束限制: 无</p> <p>取值范围: 无</p> <p>默认取值: 无</p> <p>说明 该回调函数依次包含三个参数：已上传的字节数、总字节数、已使用的时间（单位：秒）。</p>
ContentMD5	string	可选	<p>参数解释: 待上传对象数据的MD5值（经过Base64编码），提供给OBS服务端，校验数据完整性。</p> <p>约束限制: 消息体128-bit MD5值经过base64编码后得到的字符串。</p> <p>取值范围: 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。 示例：n58IG6hfM7vql4K0vnWpog==。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
WebsiteRedirectLocation	string	可选	<p>参数解释: 当桶设置了Website配置, 可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。 例如, 重定向请求到桶内另一对象: WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL: WebsiteRedirectLocation:http://www.example.com/ OBS将这个值从头域中取出, 保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向, 不支持为桶中文件夹下的对象设置重定向。 <p>取值范围: 无</p> <p>默认取值: 无</p>
ContentType	string	可选	<p>参数解释: 待上传对象的文件类型 (MIME类型)。Content-Type (MIME) 用于标识发送或接收数据的类型, 浏览器根据该参数来决定数据的打开方式。</p> <p>约束限制: 无</p> <p>取值范围: 常见的Content-Type (MIME) 列表参见如何理解Content-Type (MIME) (Node.js SDK)</p> <p>默认取值: 如果您在上传Object时未指定Content-Type, SDK会根据指定Object名称的后缀名来判定文件类型并自动填充Content-Type。</p>

参数名称	参数类型	是否必选	描述
ContentLength	int64	可选	<p>参数解释: 待上传对象数据的长度。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 单次上传对象大小范围是[0, 5GB]。• 如果需要上传超过5GB的大文件, 需要使用分段上传 <p>取值范围: [0, 5GB]。</p> <p>默认取值: 如果不传, 则sdk会自动计算上传对象数据的长度。</p>
SseKms	string	当使用SSE-KMS方式时, 必选	<p>参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p>约束限制: 当前仅支持kms。</p> <p>取值范围: 可选值: kms, 即选择SSE-KMS方式加密对象。</p> <p>默认取值: 无</p>
SseKmsKey	string	可选	<p>参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p>约束限制: 有效值支持两种格式:</p> <ol style="list-style-type: none">1. regionID:domainID(账号ID):key/key_id2. key_id <p>其中:</p> <ul style="list-style-type: none">• regionID是使用密钥所属region的ID, 可在地区和终端节点页面获取;• domainID是使用密钥所属账号的账号ID, 获取方法参见如何获取账号ID和用户ID? (Node.js SDK);• key_id是从数据加密服务创建的密钥ID, 获取方法请参见查看密钥。 <p>取值范围: 无</p> <p>默认取值:</p> <ul style="list-style-type: none">• 如果用户没有提供该头域, 那么默认的主密钥将会被使用。• 如果默认主密钥不存在, 将默认创建并使用。

参数名称	参数类型	是否必选	描述
SseC	string	当使用SSE-C方式时, 必选	<p>参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p> <p>约束限制: 仅支持AES256。</p> <p>取值范围: 可选值: AES256, 即选择SSE-C方式加密对象。</p> <p>默认取值: 无</p>
SseCKey	string	当使用SSE-C方式时, 必选	<p>参数解释: SSE-C方式下加密使用的密钥, 该密钥用于加密对象。</p> <p>约束限制: 该头域由256bit的密钥经过Base64编码得到, 示例: K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw=</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
Metadata	object	可选	<p>参数解释: 待上传对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据, 以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时, 加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为: 每个键和值的UTF-8编码中的字节总数。• 自定义元数据的key值不区分大小写, OBS统一转为小写进行存储。value值区分大小写。• 自定义元数据key-value对都必须符合US-ASCII。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。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantRead	string	可选	<p>参数解释: 被授予READ权限的账号的账号ID, 即domain_id。账号被授予READ权限后, 针对当前对象, 允许被授权指定账号有读对象和获取对象元数据的权限。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantReadAcp	string	可选	<p>参数解释: 被授予READ_ACP权限的账号的账号ID, 即domain_id。账号被授予READ_ACP权限后, 针对当前对象, 允许读对象的ACL。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantWriteAcp	string	可选	<p>参数解释: 被授予WRITE_ACP权限的账号的账号ID, 即domain_id。账号被授予WRITE_ACP权限后, 针对当前对象, 允许修改对象的ACL信息。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantFullControl	string	可选	<p>参数解释: 被授予FULL_CONTROL权限的账号的账号ID, 即 domain_id。账号被授予FULL_CONTROL权限后, 针对当前对象, 授权指定账号有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
Expires	number	可选	<p>参数解释: 表示对象的过期时间(从对象最后修改时间开始计算), 过期之后对象会被自动删除。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 设置的天数计算出的过期时间不能早于当前时间, 如10天前上传的对象, 不能设置小于10的值。 • 此字段对于每个对象仅支持上传时配置, 不支持后期通过修改元数据接口修改。 <p>取值范围: 1~(2⁶³-1), 单位: 天。</p> <p>默认取值: 无</p>

表 6-2 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限, 其他任何人都没有访问权限。
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上, 所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上, 所有人可以获取该对象内容和元数据。

常量名	原始值	说明
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	桶公共读，桶内对象公共读。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadDelivered不能应用于对象。
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	桶公共读写，桶内对象公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadWriteDelivered不能应用于对象。

常量名	原始值	说明
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	<p>设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。</p> <p>默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。</p>

表 6-3 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。</p>
ObsClient.enums.StorageClassWarm	WARM	<p>低频访问存储。</p> <p>低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。</p>
ObsClient.enums.StorageClassCold	COLD	<p>归档存储。</p> <p>归档存储适用于很少访问（平均一年访问一次）数据的业务场景。</p>

返回结果

表 6-4 返回结果

参数类型	描述
<p>表6-5</p> <p>说明</p> <p>该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释：</p> <p>接口返回信息，详见表6-5。</p>

表 6-5 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	表6-7	参数解释: 操作成功后的结果数据, 详见 表6-7 。 约束限制: 当Status大于300时为空。

表 6-6 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 6-7 PutObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
StorageClass	StorageClassType	参数解释: 对象的存储类别。 取值范围: <ul style="list-style-type: none">当对象存储类别是标准存储时, 该值为空。可选择的存储类别参见 StorageClassType。
VersionId	string	参数解释: 对象的版本号。如果桶的多版本状态为开启, 则会返回对象的版本号。
ETag	string	参数解释: 对象的ETag值, 即base64编码的128位MD5摘要。ETag是对象内容的唯一标识, 可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A, 下载对象时ETag为B, 则说明对象内容发生了变化。ETag只反映变化的内容, 而不是其元数据。上传的对象或复制操作创建的对象, 都有唯一的ETag。 约束限制: 当对象是服务端加密的对象时, ETag值不是对象的MD5值。
SseKms	string	参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。
SseKmsKey	string	参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。
SseC	string	参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。
SseCKeyMd5	string	参数解释: SSE-C方式下加密使用密钥的MD5值, 该值用于验证密钥传输过程中是否出错。 约束限制: 由密钥值经过MD5加密再经过Base64编码后得到, 示例: 4XvB3tbNTN+tIEVa0/fGaQ==

表 6-8 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

代码示例

文本上传用于直接上传字符串。您可以通过ObsClient.putObject直接上传字符串到OBS。以下代码展示了如何将指定文本对象上传至名为examplebucket桶中。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定文本对象
      Body: 'Hello OBS'
    };
    // 文本上传对象
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put object(%s) under the bucket(%s) successful!!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageClass:%s, ETag:%s", result.InterfaceResult.StorageClass, result.InterfaceResult.ETag);
      return;
    }
  } catch (error) {
    console.error(error);
  }
}
```

```
};  
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
console.log("Status: %d", result.CommonMsg.Status);  
console.log("Code: %s", result.CommonMsg.Code);  
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
  console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
  console.log(error);  
};  
};  
};  
putObject();
```

6.3 流式上传(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

本章节介绍上传单个对象到指定桶中。

接口约束

- 您必须是桶拥有者或拥有上传对象的权限，才能上传对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObject权限，如果使用桶策略则需授予PutObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 单次上传对象大小范围是[0, 5GB]。
- 如果需要上传超过5GB的大文件，需要使用[分段上传](#)。

方法定义

```
ObsClient.putObject(params)
```

请求参数

表 6-9 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my.bucket”和“my.bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>
Key	string	必选	<p>参数解释： 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制： 无</p> <p>取值范围： 长度大于0且不超过1024的字符串。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
ACL	AclType	可选	<p>参数解释: 创建对象时可指定的预定义访问策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见ACL功能介绍。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的访问策略选项参见AclType。</p> <p>默认取值: 无</p>
StorageClass	StorageClassType	可选	<p>参数解释: 创建对象时，可以加上此头域设置对象的存储类别。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的存储类别参见StorageClassType。</p> <p>默认取值: 如果未设置此头域，则对象的存储列表默认为桶存储类别。</p>
Body	string stream.Readable	可选	<p>参数解释: 待上传段的内容，支持字符串和stream.Readable对象。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 单次上传对象大小范围是[0, 5GB]。• 如果需要上传超过5GB的大文件，需要使用分段上传。• Body与SourceFile不能同时使用。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SourceFile	string	可选	<p>参数解释： 待上传对象的源文件路径。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 单次上传大小范围是[0, 5GB]。• Body与SourceFile不能同时使用。• Offset和SourceFile配套使用，用于指定上传源文件中的某一部分数据。 <p>取值范围： 无</p> <p>默认取值： 无</p>
Offset	number	可选	<p>参数解释： 源文件中某一分段的起始偏移大小。</p> <p>约束限制： Offset和SourceFile配套使用，用于指定上传源文件中的某一部分数据。</p> <p>取值范围： 非负整数，小于待上传对象的大小，单位：字节。</p> <p>默认取值： 0</p>
ProgressCallback	function	可选	<p>参数解释： 获取上传进度的回调函数。</p> <p>约束限制： 无</p> <p>取值范围： 无</p> <p>默认取值： 无</p> <p>说明 该回调函数依次包含三个参数：已上传的字节数、总字节数、已使用的时间（单位：秒）。</p>

参数名称	参数类型	是否必选	描述
ContentMD5	string	可选	<p>参数解释: 待上传对象数据的MD5值（经过Base64编码），提供给OBS服务端，校验数据完整性。</p> <p>约束限制: 消息体128-bit MD5值经过base64编码后得到的字符串。</p> <p>取值范围: 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。</p> <p>示例：n58IG6hfM7vql4K0vnWpog==。</p> <p>默认取值: 无</p>
WebsiteRedirectLocation	string	可选	<p>参数解释: 当桶设置了Website配置，可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。</p> <p>例如，重定向请求到桶内另一对象： WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL： WebsiteRedirectLocation:http://www.example.com/ OBS将这个值从头域中取出，保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头，长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向，不支持为桶中文件夹下的对象设置重定向。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
ContentType	string	可选	<p>参数解释: 待上传对象的文件类型（MIME类型）。Content-Type（MIME）用于标识发送或接收数据的类型，浏览器根据该参数来决定数据的打开方式。</p> <p>约束限制: 无</p> <p>取值范围: 常见的Content-Type（MIME）列表参见如何理解Content-Type（MIME）(Node.js SDK)</p> <p>默认取值: 如果您在上传Object时未指定Content-Type，SDK会根据指定Object名称的后缀名来判定文件类型并自动填充Content-Type。</p>
ContentLength	int64	可选	<p>参数解释: 待上传对象数据的长度。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 单次上传对象大小范围是[0, 5GB]。• 如果需要上传超过5GB的大文件，需要使用分段上传 <p>取值范围: [0, 5GB]。</p> <p>默认取值: 如果不传，则sdk会自动计算上传对象数据的长度。</p>
SseKms	string	当使用SSE-KMS方式时，必选	<p>参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p>约束限制: 当前仅支持kms。</p> <p>取值范围: 可选值：kms，即选择SSE-KMS方式加密对象。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SseKmsKey	string	可选	<p>参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p>约束限制: 有效值支持两种格式: 1. regionID:domainID(账号ID):key/key_id 2. key_id 其中:</p> <ul style="list-style-type: none">regionID是使用密钥所属region的ID, 可在地区和终端节点页面获取;domainID是使用密钥所属账号的账号ID, 获取方法参见如何获取账号ID和用户ID? (Node.js SDK);key_id是从数据加密服务创建的密钥ID, 获取方法参见查看密钥。 <p>取值范围: 无</p> <p>默认取值:</p> <ul style="list-style-type: none">如果用户没有提供该头域, 那么默认的主密钥将会被使用。如果默认主密钥不存在, 将默认创建并使用。
SseC	string	当使用SSE-C方式时, 必选	<p>参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p> <p>约束限制: 仅支持AES256。</p> <p>取值范围: 可选值: AES256, 即选择SSE-C方式加密对象。</p> <p>默认取值: 无</p>
SseCKey	string	当使用SSE-C方式时, 必选	<p>参数解释: SSE-C方式下加密使用的密钥, 该密钥用于加密对象。</p> <p>约束限制: 该头域由256bit的密钥经过Base64编码得到, 示例: K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw=</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Metadata	object	可选	<p>参数解释: 待上传对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据,以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时,加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为:每个键和值的UTF-8编码中的字节总数。 • 自定义元数据的key值不区分大小写,OBS统一转为小写进行存储。value值区分大小写。 • 自定义元数据key-value对都必须符合US-ASCII。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。 <p>取值范围: 无</p> <p>默认取值: 无</p>
GrantRead	string	可选	<p>参数解释: 被授予READ权限的账号的账号ID,即domain_id。账号被授予READ权限后,针对当前对象,允许被授权指定账号有读对象和获取对象元数据的权限。</p> <p>约束限制: 如果授权给多个租户,需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID?(Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantReadAcp	string	可选	<p>参数解释: 被授予READ_ACP权限的账号的账号ID, 即 domain_id。账号被授予READ_ACP权限后, 针对当前对象, 允许读对象的ACL。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantWriteAcp	string	可选	<p>参数解释: 被授予WRITE_ACP权限的账号的账号ID, 即 domain_id。账号被授予WRITE_ACP权限后, 针对当前对象, 允许修改对象的ACL信息。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantFullControl	string	可选	<p>参数解释: 被授予FULL_CONTROL权限的账号的账号ID, 即 domain_id。账号被授予FULL_CONTROL权限后, 针对当前对象, 授权指定账号有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Expires	number	可选	<p>参数解释: 表示对象的过期时间（从对象最后修改时间开始计算），过期之后对象会被自动删除。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 设置的天数计算出的过期时间不能早于当前时间，如10天前上传的对象，不能设置小于10的值。 此字段对于每个对象仅支持上传时配置，不支持后期通过修改元数据接口修改。 <p>取值范围: 1~(2⁶³-1)，单位：天。</p> <p>默认取值: 无</p>

表 6-10 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限，其他任何人都没有访问权限。
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上，所有人可以获得该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上，所有人可以获得该对象内容和元数据。
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获得该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获得该对象内容和元数据。

常量名	原始值	说明
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	<p>桶公共读，桶内对象公共读。</p> <p>设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。</p> <p>说明 AclPublicReadDelivered不能应用于对象。</p>
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	<p>桶公共读写，桶内对象公共读写。</p> <p>设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。</p> <p>说明 AclPublicReadWriteDelivered不能应用于对象。</p>
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	<p>设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。</p> <p>默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。</p>

表 6-11 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

返回结果

表 6-12 返回结果

参数类型	描述
<p>表6-13</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表6-13。</p>

表 6-13 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	<p>参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICommonMsg。</p>
InterfaceResult	表6-15	<p>参数解释： 操作成功后的结果数据，详见表6-15。</p> <p>约束限制： 当Status大于300时为空。</p>

表 6-14 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 6-15 PutObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
StorageClasses	StorageClassType	参数解释: 对象的存储类别。 取值范围: <ul style="list-style-type: none">当对象存储类别是标准存储时，该值为空。可选择的存储类别参见StorageClassType。
VersionId	string	参数解释: 对象的版本号。如果桶的多版本状态为开启，则会返回对象的版本号。

参数名称	参数类型	描述
ETag	string	<p>参数解释: 对象的ETag值，即base64编码的128位MD5摘要。ETag是对象内容的唯一标识，可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A，下载对象时ETag为B，则说明对象内容发生了变化。ETag只反映变化的内容，而不是其元数据。上传的对象或复制操作创建的对象，都有唯一的ETag。</p> <p>约束限制: 当对象是服务端加密的对象时，ETag值不是对象的MD5值。</p>
SseKms	string	<p>参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p>
SseKmsKey	string	<p>参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。</p>
SseC	string	<p>参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p>
SseCKeyMd5	string	<p>参数解释: SSE-C方式下加密使用密钥的MD5值，该值用于验证密钥传输过程中是否出错。</p> <p>约束限制: 由密钥值经过MD5加密再经过Base64编码后得到，示例：4XvB3tbNTN+tIEVa0/fGaQ==</p>

表 6-16 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。</p>
ObsClient.enums.StorageClassWarm	WARM	<p>低频访问存储。</p> <p>低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。</p>

常量名	原始值	说明
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

代码示例一：上传网络流

流式上传使用stream.Readable作为对象的数据源。以下代码展示了如何通过网络流上传对象至名为examplebucket桶中。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AK/SK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getNetWorkStream() {
  // 定义网络流URL
  const url = 'https://www.example.com';
  // 引入http库、https库
  const http = require('http');
  const https = require('https');
  // 根据URL选择对应的请求库
  const request = url.startsWith('http') ? http : https;
  return new Promise((resolve, reject) => {
    request.get(url, res => {
      if (res.statusCode === 200) {
        resolve(res);
      } else {
        reject(res);
      }
    });
  });
};

async function putObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定网络流
      Body: await getNetWorkStream()
    };
    // 网络流上传对象
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put object(%s) under the bucket(%s) successful!!!", params.Key, params.Bucket);
    }
  }
}
```



```
console.log("RequestId: %s", result.CommonMsg.RequestId);
console.log("StorageClass:%s, ETag:%s", result.InterfaceResult.StorageClass, result.InterfaceResult.ETag);
return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
console.log(error);
};
};
};

putObject();
```

代码示例二：上传文件流

本示例用于文件流上传至名为examplebucket桶中。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
// 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露
// 风险。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
// security_token: process.env.SECURITY_TOKEN,
// endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getFileStream() {
const fs = require('fs');
// 填写本地文件的完整路径, 从本地文件中读取数据流。
// 如果本地文件的完整路径中未指定本地路径, 则默认从示例程序所属项目对应本地路径中上传文件
return fs.createReadStream('D:\\localpath\\examplefile.txt');
}

async function putObject() {
try {
const params = {
// 指定存储桶名称
Bucket: "examplebucket",
// 指定对象名, 此处以 example/objectname 为例
Key: "example/objectname",
// 指定文件流
Body: await getFileStream()
};
// 文件流上传对象
const result = await obsClient.putObject(params);
if (result.CommonMsg.Status <= 300) {
console.log("Put bucket(%s) successful!", params.Bucket);
console.log("RequestId: %s", result.CommonMsg.RequestId);
return;
}
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
}
```

```
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};
putObject();
```

6.4 文件上传(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

本章节介绍上传单个对象到指定桶中。

接口约束

- 您必须是桶拥有者或拥有上传对象的权限，才能上传对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObject权限，如果使用桶策略则需授予PutObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 单次上传对象大小范围是[0, 5GB]。
- 如果需要上传超过5GB的大文件，需要使用[分段上传](#)。

方法定义

```
ObsClient.putObject(params)
```

请求参数

表 6-17 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my.bucket”和“my.bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>
Key	string	必选	<p>参数解释： 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制： 无</p> <p>取值范围： 长度大于0且不超过1024的字符串。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
ACL	AclType	可选	<p>参数解释: 创建对象时可指定的预定义访问策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见ACL功能介绍。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的访问策略选项参见AclType。</p> <p>默认取值: 无</p>
StorageClass	StorageClassType	可选	<p>参数解释: 创建对象时，可以加上此头域设置对象的存储类别。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的存储类别参见StorageClassType。</p> <p>默认取值: 如果未设置此头域，则对象的存储列表默认为桶存储类别。</p>
Body	string stream.Readable	可选	<p>参数解释: 待上传段的内容，支持字符串和stream.Readable对象。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 单次上传对象大小范围是[0, 5GB]。 • 如果需要上传超过5GB的大文件，需要使用分段上传。 • Body与SourceFile不能同时使用。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SourceFile	string	可选	<p>参数解释: 待上传对象的源文件路径。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 单次上传大小范围是[0, 5GB]。• Body与SourceFile不能同时使用。• Offset和SourceFile配套使用，用于指定上传源文件中的某一部分数据。 <p>取值范围: 无</p> <p>默认取值: 无</p>
Offset	number	可选	<p>参数解释: 源文件中某一分段的起始偏移大小。</p> <p>约束限制: Offset和SourceFile配套使用，用于指定上传源文件中的某一部分数据。</p> <p>取值范围: 非负整数，小于待上传对象的大小，单位：字节。</p> <p>默认取值: 0</p>
ProgressCallback	function	可选	<p>参数解释: 获取上传进度的回调函数。</p> <p>约束限制: 无</p> <p>取值范围: 无</p> <p>默认取值: 无</p> <p>说明 该回调函数依次包含三个参数：已上传的字节数、总字节数、已使用的时间（单位：秒）。</p>

参数名称	参数类型	是否必选	描述
ContentMD5	string	可选	<p>参数解释: 待上传对象数据的MD5值（经过Base64编码），提供给OBS服务端，校验数据完整性。</p> <p>约束限制: 消息体128-bit MD5值经过base64编码后得到的字符串。</p> <p>取值范围: 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。</p> <p>示例：n58IG6hfM7vql4K0vnWpog==。</p> <p>默认取值: 无</p>
WebsiteRedirectLocation	string	可选	<p>参数解释: 当桶设置了Website配置，可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。</p> <p>例如，重定向请求到桶内另一对象： WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL： WebsiteRedirectLocation:http://www.example.com/ OBS将这个值从头域中取出，保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头，长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向，不支持为桶中文件夹下的对象设置重定向。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
ContentType	string	可选	<p>参数解释: 待上传对象的文件类型（MIME类型）。Content-Type（MIME）用于标识发送或接收数据的类型，浏览器根据该参数来决定数据的打开方式。</p> <p>约束限制: 无</p> <p>取值范围: 常见的Content-Type（MIME）列表参见如何理解Content-Type（MIME）(Node.js SDK)</p> <p>默认取值: 如果您在上传Object时未指定Content-Type，SDK会根据指定Object名称的后缀名来判定文件类型并自动填充Content-Type。</p>
ContentLength	int64	可选	<p>参数解释: 待上传对象数据的长度。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 单次上传对象大小范围是[0, 5GB]。• 如果需要上传超过5GB的大文件，需要使用分段上传 <p>取值范围: [0, 5GB]。</p> <p>默认取值: 如果不传，则sdk会自动计算上传对象数据的长度。</p>
SseKms	string	当使用SSE-KMS方式时，必选	<p>参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p>约束限制: 当前仅支持kms。</p> <p>取值范围: 可选值：kms，即选择SSE-KMS方式加密对象。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SseKmsKey	string	可选	<p>参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p>约束限制: 有效值支持两种格式: 1. regionID:domainID(账号ID):key/key_id 2. key_id 其中:</p> <ul style="list-style-type: none"> regionID是使用密钥所属region的ID, 可在地区和终端节点页面获取; domainID是使用密钥所属账号的账号ID, 获取方法参见如何获取账号ID和用户ID? (Node.js SDK); key_id是从数据加密服务创建的密钥ID, 获取方法请参见查看密钥。 <p>取值范围: 无</p> <p>默认取值:</p> <ul style="list-style-type: none"> 如果用户没有提供该头域, 那么默认的主密钥将会被使用。 如果默认主密钥不存在, 将默认创建并使用。
SseC	string	当使用SSE-C方式时, 必选	<p>参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p> <p>约束限制: 仅支持AES256。</p> <p>取值范围: 可选值: AES256, 即选择SSE-C方式加密对象。</p> <p>默认取值: 无</p>
SseCKey	string	当使用SSE-C方式时, 必选	<p>参数解释: SSE-C方式下加密使用的密钥, 该密钥用于加密对象。</p> <p>约束限制: 该头域由256bit的密钥经过Base64编码得到, 示例: K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw=</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Metadata	object	可选	<p>参数解释: 待上传对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据,以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时,加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为:每个键和值的UTF-8编码中的字节总数。• 自定义元数据的key值不区分大小写,OBS统一转为小写进行存储。value值区分大小写。• 自定义元数据key-value对都必须符合US-ASCII。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。 <p>取值范围: 无</p> <p>默认取值: 无</p>
GrantRead	string	可选	<p>参数解释: 被授予READ权限的账号的账号ID,即domain_id。账号被授予READ权限后,针对当前对象,允许被授权指定账号有读对象和获取对象元数据的权限。</p> <p>约束限制: 如果授权给多个租户,需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID?(Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantReadAcp	string	可选	<p>参数解释: 被授予READ_ACP权限的账号的账号ID, 即 domain_id。账号被授予READ_ACP权限后, 针对当前对象, 允许读对象的ACL。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantWriteAcp	string	可选	<p>参数解释: 被授予WRITE_ACP权限的账号的账号ID, 即 domain_id。账号被授予WRITE_ACP权限后, 针对当前对象, 允许修改对象的ACL信息。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantFullControl	string	可选	<p>参数解释: 被授予FULL_CONTROL权限的账号的账号ID, 即 domain_id。账号被授予FULL_CONTROL权限后, 针对当前对象, 授权指定账号有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Expires	number	可选	<p>参数解释: 表示对象的过期时间（从对象最后修改时间开始计算），过期之后对象会被自动删除。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 设置的天数计算出的过期时间不能早于当前时间，如10天前上传的对象，不能设置小于10的值。 • 此字段对于每个对象仅支持上传时配置，不支持后期通过修改元数据接口修改。 <p>取值范围: 1~(2⁶³-1)，单位：天。</p> <p>默认取值: 无</p>

表 6-18 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限，其他任何人都没有访问权限。
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上，所有人可以获得该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上，所有人可以获得该对象内容和元数据。
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获得该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获得该对象内容和元数据。

常量名	原始值	说明
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	桶公共读，桶内对象公共读。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadDelivered不能应用于对象。
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	桶公共读写，桶内对象公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadWriteDelivered不能应用于对象。
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。 默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。

表 6-19 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

返回结果

表 6-20 返回结果

参数类型	描述
<p>表6-21</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表6-21。</p>

表 6-21 Response

参数名称	参数类型	描述
CommonMsg	ICCommonMsg	<p>参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICCommonMsg。</p>
InterfaceResult	表6-23	<p>参数解释： 操作成功后的结果数据，详见表6-23。</p> <p>约束限制： 当Status大于300时为空。</p>

表 6-22 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 6-23 PutObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
StorageClasses	StorageClassType	参数解释: 对象的存储类别。 取值范围: <ul style="list-style-type: none"> 当对象存储类别是标准存储时，该值为空。 可选择的存储类别参见StorageClassType。
VersionId	string	参数解释: 对象的版本号。如果桶的多版本状态为开启，则会返回对象的版本号。

参数名称	参数类型	描述
ETag	string	<p>参数解释: 对象的ETag值，即base64编码的128位MD5摘要。ETag是对象内容的唯一标识，可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A，下载对象时ETag为B，则说明对象内容发生了变化。ETag只反映变化的内容，而不是其元数据。上传的对象或复制操作创建的对象，都有唯一的ETag。</p> <p>约束限制: 当对象是服务端加密的对象时，ETag值不是对象的MD5值。</p>
SseKms	string	<p>参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p>
SseKmsKey	string	<p>参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。</p>
SseC	string	<p>参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p>
SseCKeyMd5	string	<p>参数解释: SSE-C方式下加密使用密钥的MD5值，该值用于验证密钥传输过程中是否出错。</p> <p>约束限制: 由密钥值经过MD5加密再经过Base64编码后得到，示例：4XvB3tbNTN+tIEVa0/fGaQ==</p>

表 6-24 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。</p>
ObsClient.enums.StorageClassWarm	WARM	<p>低频访问存储。</p> <p>低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。</p>

常量名	原始值	说明
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

代码示例

文件上传使用本地文件作为对象的数据源。以下代码展示了如何通过文件上传至名为 examplebucket 桶中。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // localfile为待上传的本地文件路径，需要指定到具体的文件名
      SourceFile: 'localfile'
    };
    // 文件上传
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

putObject();
```


6.5 创建文件夹(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

本章节介绍上传文件夹对象到指定桶中，来实现新建一个文件夹，从而更方便的对存储在对象桶中的数据进行分类管理。

接口约束

- 您必须是桶拥有者或拥有上传对象的权限，才能上传对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObject权限，如果使用桶策略则需授予PutObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 创建文件夹本质上来说是创建了一个大小为0且对象名以“/”结尾的对象。

方法定义

```
ObsClient.putObject(params)
```

请求参数

表 6-25 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.bucket-”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>
Key	string	必选	<p>参数解释： 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制： 无</p> <p>取值范围： 长度大于0且不超过1024的字符串。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
ACL	AclType	可选	<p>参数解释: 创建对象时可指定的预定义访问策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见ACL功能介绍。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的访问策略选项参见AclType。</p> <p>默认取值: 无</p>
StorageClass	StorageClassType	可选	<p>参数解释: 创建对象时，可以加上此头域设置对象的存储类别。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的存储类别参见StorageClassType。</p> <p>默认取值: 如果未设置此头域，则对象的存储列表默认为桶存储类别。</p>
Body	string stream.Readable	可选	<p>参数解释: 待上传段的内容，支持字符串和stream.Readable对象。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 单次上传对象大小范围是[0, 5GB]。 • 如果需要上传超过5GB的大文件，需要使用分段上传。 • Body与SourceFile不能同时使用。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SourceFile	string	可选	<p>参数解释: 待上传对象的源文件路径。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 单次上传大小范围是[0, 5GB]。• Body与SourceFile不能同时使用。• Offset和SourceFile配套使用，用于指定上传源文件中的某一部分数据。 <p>取值范围: 无</p> <p>默认取值: 无</p>
Offset	number	可选	<p>参数解释: 源文件中某一分段的起始偏移大小。</p> <p>约束限制: Offset和SourceFile配套使用，用于指定上传源文件中的某一部分数据。</p> <p>取值范围: 非负整数，小于待上传对象的大小，单位：字节。</p> <p>默认取值: 0</p>
ProgressCallback	function	可选	<p>参数解释: 获取上传进度的回调函数。</p> <p>约束限制: 无</p> <p>取值范围: 无</p> <p>默认取值: 无</p> <p>说明 该回调函数依次包含三个参数：已上传的字节数、总字节数、已使用的时间（单位：秒）。</p>

参数名称	参数类型	是否必选	描述
ContentMD5	string	可选	<p>参数解释: 待上传对象数据的MD5值（经过Base64编码），提供给OBS服务端，校验数据完整性。</p> <p>约束限制: 消息体128-bit MD5值经过base64编码后得到的字符串。</p> <p>取值范围: 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。</p> <p>示例：n58IG6hfM7vql4K0vnWpog==。</p> <p>默认取值: 无</p>
WebsiteRedirectLocation	string	可选	<p>参数解释: 当桶设置了Website配置，可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。</p> <p>例如，重定向请求到桶内另一对象： WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL： WebsiteRedirectLocation:http://www.example.com/ OBS将这个值从头域中取出，保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头，长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向，不支持为桶中文件夹下的对象设置重定向。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
ContentType	string	可选	<p>参数解释: 待上传对象的文件类型（MIME类型）。Content-Type（MIME）用于标识发送或接收数据的类型，浏览器根据该参数来决定数据的打开方式。</p> <p>约束限制: 无</p> <p>取值范围: 常见的Content-Type（MIME）列表参见如何理解Content-Type（MIME）(Node.js SDK)</p> <p>默认取值: 如果您在上传Object时未指定Content-Type，SDK会根据指定Object名称的后缀名来判定文件类型并自动填充Content-Type。</p>
ContentLength	int64	可选	<p>参数解释: 待上传对象数据的长度。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 单次上传对象大小范围是[0, 5GB]。 • 如果需要上传超过5GB的大文件，需要使用分段上传 <p>取值范围: [0, 5GB]。</p> <p>默认取值: 如果不传，则sdk会自动计算上传对象数据的长度。</p>
SseKms	string	当使用SSE-KMS方式时，必选	<p>参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p>约束限制: 当前仅支持kms。</p> <p>取值范围: 可选值：kms，即选择SSE-KMS方式加密对象。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SseKmsKey	string	可选	<p>参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p>约束限制: 有效值支持两种格式: 1. regionID:domainID(账号ID):key/key_id 2. key_id 其中:</p> <ul style="list-style-type: none"> regionID是使用密钥所属region的ID, 可在地区和终端节点页面获取; domainID是使用密钥所属账号的账号ID, 获取方法参见如何获取账号ID和用户ID? (Node.js SDK); key_id是从数据加密服务创建的密钥ID, 获取方法参见查看密钥。 <p>取值范围: 无</p> <p>默认取值:</p> <ul style="list-style-type: none"> 如果用户没有提供该头域, 那么默认的主密钥将会被使用。 如果默认主密钥不存在, 将默认创建并使用。
SseC	string	当使用SSE-C方式时, 必选	<p>参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p> <p>约束限制: 仅支持AES256。</p> <p>取值范围: 可选值: AES256, 即选择SSE-C方式加密对象。</p> <p>默认取值: 无</p>
SseCKey	string	当使用SSE-C方式时, 必选	<p>参数解释: SSE-C方式下加密使用的密钥, 该密钥用于加密对象。</p> <p>约束限制: 该头域由256bit的密钥经过Base64编码得到, 示例: K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw=</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Metadata	object	可选	<p>参数解释: 待上传对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据,以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时,加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为:每个键和值的UTF-8编码中的字节总数。• 自定义元数据的key值不区分大小写,OBS统一转为小写进行存储。value值区分大小写。• 自定义元数据key-value对都必须符合US-ASCII。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。 <p>取值范围: 无</p> <p>默认取值: 无</p>
GrantRead	string	可选	<p>参数解释: 被授予READ权限的账号的账号ID,即domain_id。账号被授予READ权限后,针对当前对象,允许被授权指定账号有读对象和获取对象元数据的权限。</p> <p>约束限制: 如果授权给多个租户,需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID?(Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantReadAcp	string	可选	<p>参数解释: 被授予READ_ACP权限的账号的账号ID, 即 domain_id。账号被授予READ_ACP权限后, 针对当前对象, 允许读对象的ACL。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantWriteAcp	string	可选	<p>参数解释: 被授予WRITE_ACP权限的账号的账号ID, 即 domain_id。账号被授予WRITE_ACP权限后, 针对当前对象, 允许修改对象的ACL信息。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantFullControl	string	可选	<p>参数解释: 被授予FULL_CONTROL权限的账号的账号ID, 即 domain_id。账号被授予FULL_CONTROL权限后, 针对当前对象, 授权指定账号有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Expires	number	可选	<p>参数解释: 表示对象的过期时间（从对象最后修改时间开始计算），过期之后对象会被自动删除。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 设置的天数计算出的过期时间不能早于当前时间，如10天前上传的对象，不能设置小于10的值。 • 此字段对于每个对象仅支持上传时配置，不支持后期通过修改元数据接口修改。 <p>取值范围: 1~(2⁶³-1)，单位：天。</p> <p>默认取值: 无</p>

表 6-26 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限，其他任何人都没有访问权限。
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上，所有人可以获得该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上，所有人可以获得该对象内容和元数据。
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获得该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获得该对象内容和元数据。

常量名	原始值	说明
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	桶公共读，桶内对象公共读。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadDelivered不能应用于对象。
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	桶公共读写，桶内对象公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadWriteDelivered不能应用于对象。
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。 默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。

表 6-27 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

返回结果

表 6-28 返回结果

参数类型	描述
<p>表6-29</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表6-29。</p>

表 6-29 Response

参数名称	参数类型	描述
CommonMsg	ICCommonMsg	<p>参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICCommonMsg。</p>
InterfaceResult	表6-31	<p>参数解释： 操作成功后的结果数据，详见表6-31。</p> <p>约束限制： 当Status大于300时为空。</p>

表 6-30 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 6-31 PutObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
StorageClasses	StorageClassType	参数解释: 对象的存储类别。 取值范围: <ul style="list-style-type: none"> 当对象存储类别是标准存储时，该值为空。 可选择的存储类别参见StorageClassType。
VersionId	string	参数解释: 对象的版本号。如果桶的多版本状态为开启，则会返回对象的版本号。

参数名称	参数类型	描述
ETag	string	<p>参数解释: 对象的ETag值，即base64编码的128位MD5摘要。ETag是对象内容的唯一标识，可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A，下载对象时ETag为B，则说明对象内容发生了变化。ETag只反映变化的内容，而不是其元数据。上传的对象或复制操作创建的对象，都有唯一的ETag。</p> <p>约束限制: 当对象是服务端加密的对象时，ETag值不是对象的MD5值。</p>
SseKms	string	<p>参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p>
SseKmsKey	string	<p>参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。</p>
SseC	string	<p>参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p>
SseCKeyMd5	string	<p>参数解释: SSE-C方式下加密使用密钥的MD5值，该值用于验证密钥传输过程中是否出错。</p> <p>约束限制: 由密钥值经过MD5加密再经过Base64编码后得到，示例：4XvB3tbNTN+tIEVa0/fGaQ==</p>

表 6-32 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。</p>
ObsClient.enums.StorageClassWarm	WARM	<p>低频访问存储。</p> <p>低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。</p>

常量名	原始值	说明
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

代码示例

OBS本身是没有文件夹的概念，桶中存储的元素只有对象。创建文件夹实际上是创建了一个大小为0且对象名以“/”结尾的对象，这类对象与其他对象无任何差异，可以进行下载、删除等操作，只是OBS控制台会将这类以“/”结尾的对象以文件夹的方式展示。本示例用于创建名为parent_directory的文件夹对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定文件夹，此处以 parent_directory 为例
      Key: "parent_directory/",
    };
    // 创建文件夹
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

putObject();
```

📖 说明

- 创建文件夹本质上来说是创建了一个大小为0且对象名以“/”结尾的对象。

6.6 设置对象元数据(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

元数据（Metadata）为描述对象属性的信息，是一组名称值对，用作对象管理的一部分。

当前仅支持系统定义的元数据。系统定义的元数据又分为两种类别：系统控制和用户控制。

对指定桶中的对象发送HEAD请求，获取对象的元数据信息。

接口约束

- 您必须是桶拥有者或拥有获取对象元数据的权限，才能获取对象元数据。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:GetObject权限，如果使用桶策略则需授予GetObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 拥有对象读权限的用户可以执行HEAD操作命令获取对象元数据，返回信息包含对象的元数据信息。

方法定义

```
ObsClient.setObjectMetadata(params)
```


请求参数

表 6-33 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p>
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
MetadataDirective	MetadataDirectiveType	必选	<p>参数解释: 复制源对象属性的策略。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的对象属性选项参见 MetadataDirectiveType。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如： G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
CacheControl	string	可选	<p>参数解释: 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Cache-Control的取值。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
ContentDisposition	string	可选	参数解释: 响应中的Content-Disposition头。指定对象被下载时的名称。 约束限制: 无 取值范围: 参见HTTP标准头域Content-Disposition的取值。 默认取值: 无
ContentEncoding	string	可选	参数解释: 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。 约束限制: 无 取值范围: 参见HTTP标准头域Content-Encoding的取值。 默认取值: 无
ContentLanguage	string	可选	参数解释: 响应中的Content-Language头。指定对象被下载时的内容语言格式。 约束限制: 无 取值范围: 参见HTTP标准头域Content-Language的取值。 默认取值: 无

参数名称	参数类型	是否必选	描述
ContentType	string	可选	<p>参数解释: 响应中的Content-Type头。指定对象被下载时的文件类型。</p> <p>约束限制: 无</p> <p>取值范围: 常见的Content-Type (MIME) 列表参见如何理解Content-Type (MIME) (Node.js SDK)</p> <p>默认取值: 无</p>
Expires	string	可选	<p>参数解释: 响应中的Expires头。指定对象被下载时的网页的缓存过期时间。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Expires的取值。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
WebsiteRedirectLocation	string	可选	<p>参数解释: 当桶设置了Website配置,可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。 例如,重定向请求到桶内另一对象: WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL: WebsiteRedirectLocation:http://www.example.com/ OBS将这个值从头域中取出,保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 必须以“/”、“http://”或“https://”开头,长度不超过2KB。 • OBS仅支持为桶根目录下的对象设置重定向,不支持为桶中文件夹下的对象设置重定向。 <p>取值范围: 无</p> <p>默认取值: 无</p>
StorageClass	StorageClassType	可选	<p>参数解释: 指定对象的存储类别。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的存储类别参见StorageClassType。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Metadata	object	可选	<p>参数解释: 对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据，以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时，加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为：每个键和值的UTF-8 编码中的字节总数。 • 自定义元数据的key值不区分大小写，OBS统一转为小写进行存储。value值区分大小写。 • 自定义元数据key-value对都必须符合US-ASCII。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。 <p>取值范围: 无</p> <p>默认取值: 无</p>

表 6-34 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。

常量名	原始值	说明
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

返回结果

表 6-35 返回结果

参数类型	描述
<p>表6-36</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表6-36。</p>

表 6-36 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	<p>参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICommonMsg。</p>
InterfaceResult	表6-38	<p>参数解释： 操作成功后的结果数据，详见表6-38。</p> <p>约束限制： 当Status大于300时为空。</p>

表 6-37 ICommonMsg

参数名称	参数类型	描述
Status	number	<p>参数解释： OBS服务端返回的HTTP状态码。</p> <p>取值范围： 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。</p>
Code	string	<p>参数解释： OBS服务端返回的错误码。</p>

Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 6-38 SetObjectMetadataOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
MetadataDirective	Metadata Directive Type	参数解释: 复制源对象属性的策略。 取值范围: 可选择的对象属性选项参见 MetadataDirectiveType 。
CacheControl	string	参数解释: 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。
ContentDisposition	string	参数解释: 响应中的Content-Disposition头。指定对象被下载时的名称。
ContentEncoding	string	参数解释: 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。
ContentLanguage	string	参数解释: 响应中的Content-Language头。指定对象被下载时的内容语言格式。
ContentType	string	参数解释: 响应中的Content-Type头。指定对象被下载时的文件类型。

参数名称	参数类型	描述
Expires	string	参数解释: 响应中的Expires头。指定对象被下载时的网页的缓存过期时间。
WebsiteRedirectLocation	string	参数解释: 当桶设置了Website配置，可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。 例如，重定向请求到桶内另一对象： WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL： WebsiteRedirectLocation:http://www.example.com/ OBS将这个值从头域中取出，保存在对象的元数据“WebsiteRedirectLocation”中。 约束限制: <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头，长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向，不支持为桶中文件夹下的对象设置重定向。
StorageClass	StorageClassType	参数解释: 复制时设置对象的存储类别。如果未设置此头域，则以桶的默认存储类别作为对象的存储类别。 取值范围: 可选择的存储类别参见 StorageClassType 。

参数名称	参数类型	描述
Metadata	object	<p>参数解释: 对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据，以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时，加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为：每个键和值的UTF-8 编码中的字节总数。 自定义元数据的key值不区分大小写，OBS统一转为小写进行存储。value值区分大小写。 自定义元数据key-value对都必须符合US-ASCII。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。

表 6-39 MetadataDirectiveType

常量名	原始值	说明
ObsClient.enums.CopyMetadata	COPY	复制对象时，复制源对象属性。 须知 该值仅在 复制对象(Node.js SDK) 接口中使用
ObsClient.enums.ReplaceMetadata	REPLACE	使用当前请求中携带的头域完整替换，未指定的元数据会被删除。
ObsClient.enums.ReplaceNewMetadata	REPLACE_NEW	对于已经存在值的元数据进行替换，不存在值的元数据进行赋值，未指定的元数据保持不变。 须知 该值仅在 设置对象元数据(Node.js SDK) 接口中使用

表 6-40 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。

常量名	原始值	说明
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

代码示例

本示例用于设置examplebucket桶中的example/objectname对象元数据，并且将对象MIME类型设置成image/jpeg，对象存储类型指定为归档存储，并设置自定义元数据。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AK/SK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setObjectMetadata() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定对象MIME类型，这里以image/jpeg为例
      ContentType: "image/jpeg",
      // 指定对象存储类型，这里以 obsClient.enums.StorageClassCold 为例
      StorageClass: obsClient.enums.StorageClassCold,
      // 指定自定义元数据
      Metadata: { "property1": "property-value1", "property2": "property-value2" }
    };
    // 设置对象元数据
    const result = await obsClient.setObjectMetadata(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set Object(%s)'s metadata successful with bucket(%s)!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
  }
}
```

```
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};
setObjectMetadata();
```

6.7 上传对象时设置指定对象生命周期(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

上传对象或者初始化分段上传任务时，您可以使用**Expires**参数直接指定对象的过期时间。此方式仅支持设置以天为单位的对象过期时间，过期后的对象会被OBS服务端自动清理，其优先级高于桶生命周期规则。

代码示例一：上传对象

上传对象时，您可以直接指定对象的过期时间。代码示例如下：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // localfile为待上传的本地文件路径，需要指定到具体的文件名
      SourceFile: 'localfile',
      // 指定对象的过期时间，此处以30为例，单位是天
      Expires: 30
    };
    // 文件上传
    const result = await obsClient.putObject(params);
```

```
    if (result.CommonMsg.Status <= 300) {
      console.log("Put bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

putObject()
```

代码示例二：初始化分段上传任务

初始化分段上传任务时，您可以直接指定对象的过期时间。代码示例如下：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function initiateMultipartUpload() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例。
      Key: "example/objectname",
      // 指定对象的过期时间，此处以30为例，单位是天
      Expires: 30
    };
    // 初始化上传任务
    const result = await obsClient.initiateMultipartUpload(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Initiate multipart upload successfull with bucket(%s) and object(%s)!", params.Bucket, params.Key);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("UploadId: %s", result.InterfaceResult.UploadId);
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};
```

```
    } catch (error) {  
      console.log("An Exception was found, which means the client encountered an internal problem when  
      attempting to communicate with OBS, for example, the client was unable to access the network.");  
      console.log(error);  
    };  
  };  
};  
  
initiateMultipartUpload();
```

6.8 追加上传(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

您可以根据需要上传文件或者文件夹至已有的OBS桶。待上传的文件可以是任何类型：文本文件、图片、视频等。

追加写对象操作是指在指定对象尾追加上传数据，不存在相同对象键值的对象则创建新对象。

每次追加上传都会更新该对象的最后修改时间。

接口约束

- 您必须是桶拥有者或拥有上传对象的权限，才能上传对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObject权限，如果使用桶策略则需授予PutObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 同一个桶中存储的对象名必须是唯一的。
- 启用服务端加密SSE-C方式时，追加上传和初始化段一样，设置诸如x-obs-server-side-encryption之类的请求Header，后续追加上传也必须携带。
- 启用服务端加密SSE-KMS方式时，有且只有第一次上传且桶内不存在同名对象时，才设置诸如x-obs-server-side-encryption之类的请求Header，后续追加上传不携带，会继承之前的设置。
- 每次追加上传的长度不能超过对象长度上限5G的限制。
- 每个Appendable对象追加上传次数最多为10000次。
- 如果对象存储类别为COLD（归档存储），则不能调用该接口。
- 如果桶设置了跨区域复制配置，则不能调用该接口。
- 并行文件系统不支持追加上传对象。
- ObsClient.putObject上传的对象可覆盖ObsClient.appendObject上传的对象，覆盖后对象变为普通对象，不可再进行追加上传。
- 第一次调用追加上传时，如果已存在同名的普通对象，则会抛出异常（HTTP状态码为409）。
- 追加上传返回的ETag是当次追加数据内容的ETag，不是完整对象的ETag。

方法定义

ObsClient.appendObject(params)

请求参数

表 6-41 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。同一用户在同一区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
Position	number	必选	<p>参数解释: 追加写位置。</p> <p>约束限制: 需要追加写的对象首次上传时必须指定为0，下次追加需要填写的值为服务端本次上传成功返回消息的 NextAppendPosition。</p> <p>取值范围: 0~(2⁶³-1)，单位：字节。</p> <p>默认取值: 0</p>
Body	string stream.Readable	可选	<p>参数解释: 待上传段的内容，支持字符串和 stream.Readable对象。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 单次上传对象大小范围是[0, 5GB]。• 如果需要上传超过5GB的大文件，需要使用分段上传。• Body与SourceFile不能同时使用。• 当Body与SourceFile都为空时，上传对象的大小为0字节。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SourceFile	string	可选	<p>参数解释: 待上传对象的源文件路径。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 单次上传大小范围是[0, 5GB]。 • Body与SourceFile不能同时使用。 • 当Body与SourceFile都为空时，上传对象的大小为0字节。 • Offset和SourceFile配套使用，用于指定上传源文件中的某一部分数据。 <p>取值范围: 无</p> <p>默认取值: 无</p>
Offset	number	可选	<p>参数解释: 源文件中某一分段的起始偏移大小。</p> <p>约束限制: Offset和SourceFile配套使用，用于指定上传源文件中的某一部分数据。</p> <p>取值范围: 非负整数，小于待上传对象的大小，单位：字节。</p> <p>默认取值: 0</p>
ProgressCallback	function	可选	<p>参数解释: 获取上传进度的回调函数。</p> <p>说明 该回调函数依次包含三个参数：已上传的字节数、总字节数、已使用的时间（单位：秒）。</p> <p>约束限制: 无</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
ContentMD5	string	可选	<p>参数解释: 待上传对象数据的MD5值（经过Base64编码），提供给OBS服务端，校验数据完整性。</p> <p>约束限制: 消息体128-bit MD5值经过base64编码后得到的字符串。</p> <p>取值范围: 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。</p> <p>示例： n58lG6hfM7vql4K0vnWpog==。</p> <p>默认取值: 无</p>
ACL	AclType	可选	<p>参数解释: 创建对象时可指定的预定义访问策略。有关访问控制列表（Access Control List, ACL）功能的详细信息可参见ACL 功能介绍。</p> <p>约束限制: 仅第一次追加上传时可设置，后续追加上传时，系统默认与第一次追加上传的配置保持一致。</p> <p>取值范围: 可选择的访问策略选项参见AclType。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
WebsiteRedirectLocation	string	可选	<p>参数解释:</p> <p>当桶设置了Website配置, 可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。</p> <p>例如, 重定向请求到桶内另一对象: WebsiteRedirectLocation:/anotherPage.html</p> <p>或重定向请求到一个外部URL: WebsiteRedirectLocation:http://www.example.com/</p> <p>OBS将这个值从头域中取出, 保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 仅第一次追加上传时可设置, 后续追加上传时, 系统默认与第一次追加上传的配置保持一致。• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向, 不支持为桶中文件夹下的对象设置重定向。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
ContentType	string	可选	<p>参数解释: 待上传对象的文件类型（MIME类型）。Content-Type（MIME）用于标识发送或接收数据的类型，浏览器根据该参数来决定数据的打开方式。</p> <p>约束限制: 仅第一次追加上传时可设置，后续追加上传时，系统默认与第一次追加上传的配置保持一致。</p> <p>取值范围: 常见的Content-Type（MIME）列表参见如何理解Content-Type（MIME）(Node.js SDK)</p> <p>默认取值: 如果您在上传Object时未指定Content-Type，SDK会根据指定Object名称的后缀名来判定文件类型并自动填充Content-Type。</p>
ContentLength	int64	可选	<p>参数解释: 待上传对象数据的长度。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 单次上传对象大小范围是[0, 5GB]。• 如果需要上传超过5GB的大文件，需要使用分段上传 <p>取值范围: [0, 5GB]。</p> <p>默认取值: 如果不传，则SDK会自动计算上传对象数据的长度。</p>

参数名称	参数类型	是否必选	描述
SseKms	string	当使用SSE-KMS方式时，必选	<p>参数解释： 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 当前仅支持kms。• 仅第一次追加上传时可设置，后续追加上传时，系统默认与第一次追加上传的配置保持一致。 <p>取值范围： 可选值：kms，即选择SSE-KMS方式加密对象。</p> <p>默认取值： 无</p>
SseKmsKey	string	可选	<p>参数解释： SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p>约束限制： 仅第一次追加上传时可设置，后续追加上传时，系统默认与第一次追加上传的配置保持一致。</p> <p>取值范围： 有效值支持两种格式： 1. regionID:domainID(账号ID):key/key_id 2. key_id 其中：</p> <ul style="list-style-type: none">• regionID是使用密钥所属region的ID，可在地区和终端节点页面获取；• domainID是使用密钥所属账号的账号ID，获取方法参见如何获取账号ID和用户ID? (Node.js SDK)；• key_id是从数据加密服务创建的密钥ID，获取方法请参见查看密钥。 <p>默认取值：</p> <ul style="list-style-type: none">• 如果用户没有提供该头域，那么默认的主密钥将会被使用。• 如果默认主密钥不存在，将默认创建并使用。

参数名称	参数类型	是否必选	描述
SseC	string	当使用SSE-C方式时，必选	<p>参数解释： 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 仅支持AES256。• 仅第一次追加上传时可设置，后续追加上传时，系统默认与第一次追加上传的配置保持一致。 <p>取值范围： 可选值：AES256，即选择SSE-C方式加密对象。</p> <p>默认取值： 无</p>
SseCKey	string	当使用SSE-C方式时，必选	<p>参数解释： SSE-C方式下加密使用的密钥，该密钥用于加密对象。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVao bncnLht/rCB2o/9Cw=• 仅第一次追加上传时可设置，后续追加上传时，系统默认与第一次追加上传的配置保持一致。 <p>取值范围： 无</p> <p>默认取值： 无</p>
StorageClass	StorageClassType	可选	<p>参数解释： 创建对象时，可以加上此头域设置对象的存储类别。</p> <p>约束限制： 无</p> <p>取值范围： 可选择的存储类别参见StorageClassType。</p> <p>默认取值： 如果未设置此头域，则对象的存储列表默认为桶存储类别。</p>

参数名称	参数类型	是否必选	描述
Metadata	object	可选	<p>参数解释: 待上传对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据，以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时，加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 仅第一次追加上传有效。 • 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为：每个键和值的UTF-8 编码中的字节总数。 • 自定义元数据的key值不区分大小写，OBS统一转为小写进行存储。value值区分大小写。 • 自定义元数据key-value对都必须符合US-ASCII。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。 <p>取值范围: 无</p> <p>默认取值: 无</p>
GrantRead	string	可选	<p>参数解释: 被授予READ权限的账号的账号ID，即domain_id。账号被授予READ权限后，针对当前对象，允许被授权指定账号有读对象和获取对象元数据的权限。</p> <p>约束限制: 如果授权给多个租户，需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantReadAcp	string	可选	<p>参数解释: 被授予READ_ACP权限的账号的账号ID, 即domain_id。账号被授予READ_ACP权限后, 针对当前对象, 允许读对象的ACL。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantWriteAcp	string	可选	<p>参数解释: 被授予WRITE_ACP权限的账号的账号ID, 即domain_id。账号被授予WRITE_ACP权限后, 针对当前对象, 允许修改对象的ACL信息。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantFullControl	string	可选	<p>参数解释: 被授予FULL_CONTROL权限的账号的账号ID, 即domain_id。账号被授予FULL_CONTROL权限后, 针对当前对象, 授权指定账号有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Expires	number	可选	<p>参数解释: 表示对象的过期时间（从对象最后修改时间开始计算），过期之后对象会被自动删除。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 设置的天数计算出的过期时间不能早于当前时间，如10天前上传的对象，不能设置小于10的值。 • 此字段对于每个对象仅支持上传时配置，不支持后期通过修改元数据接口修改。 <p>取值范围: 1~(2⁶³-1)，单位：天。</p> <p>默认取值: 无</p>

表 6-42 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限，其他任何人都没有访问权限。
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获取该对象内容和元数据。

常量名	原始值	说明
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	桶公共读，桶内对象公共读。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadDelivered不能应用于对象。
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	桶公共读写，桶内对象公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadWriteDelivered不能应用于对象。
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。 默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。

表 6-43 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

返回结果

表 6-44 返回结果

参数类型	描述
<p>表6-45</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表6-45。</p>

表 6-45 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	<p>参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICommonMsg。</p>
InterfaceResult	表6-47	<p>参数解释： 操作成功后的结果数据，详见表6-47。</p> <p>约束限制： 当Status大于300时为空。</p>

表 6-46 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 6-47 AppendObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
VersionId	string	参数解释: 对象的版本号。如果桶的多版本状态为开启，则会返回对象的版本号。

参数名称	参数类型	描述
ETag	string	参数解释: 对象的ETag值,即base64编码的128位MD5摘要。ETag是对象内容的唯一标识,可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A,下载对象时ETag为B,则说明对象内容发生了变化。ETag只反映变化的内容,而不是其元数据。上传的对象或复制操作创建的对象,都有唯一的ETag。 约束限制: 当对象是服务端加密的对象时,ETag值不是对象的MD5值。
NextAppendPosition	int64	参数解释: 下一次追加上传请求应该提供的追加写位置(position)。 约束限制: 需要追加写的对象首次上传时必须指定为0,下次追加需要填写的值为服务端本次上传成功返回消息的NextAppendPosition,也可通过ObsClient.getObjectMetadata接口获取NextAppendPosition。
SseKms	string	参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。
SseKmsKey	string	参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。
SseC	string	参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。
SseCKeyMd5	string	参数解释: SSE-C方式下加密使用密钥的MD5值,该值用于验证密钥传输过程中是否出错。 约束限制: 由密钥值经过MD5加密再经过Base64编码后得到,示例:4XvB3tbNTN+tIEVa0/fGaQ==

代码示例

追加上传可实现对同一个对象追加数据内容的功能,您可以通过ObsClient.appendObject进行追加上传。本示例用于追加上传examplebucket桶中的example/objectname对象。

```
// 引入obs库  
// 使用npm安装
```

```
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function appendObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名, 此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定追加写位置。首次上传时必须指定position为0, 下次追加需要填写的position
      Position: 0,
      // 指定待上传对象的数据流
      Body: strings.NewReader("Hello OBS"),
    };
    // 追加上传对象
    const result = await obsClient.appendObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Append object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("ETag: %s, NextAppendPosition:%d", result.InterfaceResult.ETag, result.InterfaceResult.NextAppendPosition);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

appendObject();
```

6.9 断点续传上传(Node.js SDK)

须知

开发过程中, 您有任何问题可以在github上[提交issue](#)。

功能说明

断点续传上传接口是利用[分段上传](#)特性实现的，是对分段上传的封装和加强，解决上传大文件时由于网络不稳定或程序崩溃导致上传失败的问题。

当上传大文件时，经常出现因网络不稳定或程序崩溃导致上传失败的情况。失败后再次重新上传不仅浪费资源，而且当网络不稳定时仍然有上传失败的风险。断点续传上传接口能有效地解决此类问题引起的上传失败，其原理是将待上传的文件分成若干个分段分别上传，并实时地将每段上传结果统一记录在checkpoint文件中，仅当所有分段都上传成功时返回上传成功的结果，否则在回调函数中返回错误码提醒用户再次调用接口进行重新上传（重新上传时因为有checkpoint文件记录当前的上传进度，避免重新上传所有分段，从而节省资源提高效率）。

断点续传上传接口不仅能在失败重传时节省资源提高效率，还因其对分段进行并发上传的机制能加快上传速度，能帮助用户快速完成上传业务；且其对用户透明，用户不用关心checkpoint文件的创建和删除、分段任务的切分、并发上传的实现等内部细节。

接口约束

断点续传上传接口传入的文件总大小至少要100K以上。

方法定义

```
ObsClient.uploadFile(params)
```

请求参数

表 6-48 请求参数列表

参数名称	类型	是否必选	说明
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	类型	是否必选	说明
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
UploadFile	string	可选	<p>参数解释: 待上传对象的源文件路径。</p> <p>约束限制: 无</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
PartSize	number	可选	<p>参数解释: 当前段的长度。</p> <p>约束限制: 无</p> <p>取值范围: 100KB~5GB，单位：字节。</p> <p>默认取值: 102400字节</p>

参数名称	类型	是否必选	说明
TaskNum	number	可选	参数解释: 分段上传时的最大并发数。 约束限制: 无 取值范围: 1~10000 默认取值: 1, 即不设置则默认串行上传。
ProgressCallback	function	可选	参数解释: 获取上传进度的回调函数。 说明 该回调函数依次包含三个参数: 已上传的字节数、总字节数、已使用的时间(单位: 秒)。 约束限制: 无 取值范围: 无 默认取值: 无
ResumeCallback	function	可选	参数解释: 获取取消断点续传控制参数的回调函数。 约束限制: 无 取值范围: 无 默认取值: 无 说明 <ul style="list-style-type: none">该回调函数有一个参数: 取消断点续传上传任务控制参数, 该参数有暂停断点续传上传任务和取消断点续传上传任务两个方法;可以调用取消断点续传上传任务控制参数的cancel方法来暂停断点续传上传任务。可以调用取消断点续传上传任务控制参数的abort方法来取消断点续传上传任务。

参数名称	类型	是否必选	说明
EnableCheckpoint	boolean	可选	<p>参数解释: 是否开启断点续传模式。</p> <p>约束限制: 无</p> <p>取值范围:</p> <ul style="list-style-type: none">• true: 开启断点续传模式。• false: 关闭断点续传模式, 此时断点续传上传接口退化成对分段上传的简单封装, 不会产生checkpoint文件。 <p>默认取值: false</p>
CheckpointFile	string	可选	<p>参数解释: 断点续传过程中, 会生成一个进度记录文件, 文件中会记录段的上传进度和段的相关信息。 checkpointFile参数为该记录文件的文件路径。</p> <p>约束限制: 仅在断点续传模式下有效。</p> <p>取值范围: 无</p> <p>默认取值: 当该值为空时, 默认为待上传的本地文件的同级目录。</p>
EnableCheckSum	boolean	可选	<p>参数解释: 是否校验待上传文件的内容, 如果开启该参数, 会在每次任务重新开始前对待上传文件进行校验, 验证是否与任务初始化时使用文件为同一文件。</p> <p>约束限制: 仅在断点续传模式下有效。</p> <p>取值范围:</p> <ul style="list-style-type: none">• true: 校验待上传文件的内容。• false: 不校验待上传文件的内容。 <p>默认取值: false</p>

参数名称	类型	是否必选	说明
ContentType	string	可选	<p>参数解释: 待上传对象的文件类型（MIME类型）。Content-Type（MIME）用于标识发送或接收数据的类型，浏览器根据该参数来决定数据的打开方式。</p> <p>约束限制: 无</p> <p>取值范围: 常见的Content-Type（MIME）列表参见如何理解Content-Type（MIME）(Node.js SDK)</p> <p>默认取值: 如果您在上传Object时未指定Content-Type，SDK会根据指定Object名称的后缀名来判定文件类型并自动填充Content-Type。</p>
ACL	AclType	可选	<p>参数解释: 创建对象时可指定的预定义访问策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见ACL功能介绍。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的访问策略选项参见AclType。</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
WebsiteRedirectLocation	string	可选	<p>参数解释:</p> <p>当桶设置了Website配置, 可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。</p> <p>例如, 重定向请求到桶内另一对象:</p> <p>WebsiteRedirectLocation:/anotherPage.html</p> <p>或重定向请求到一个外部URL:</p> <p>WebsiteRedirectLocation:http://www.example.com/</p> <p>OBS将这个值从头域中取出, 保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向, 不支持为桶中文件夹下的对象设置重定向。 <p>取值范围:</p> <p>无</p> <p>默认取值:</p> <p>无</p>

参数名称	类型	是否必选	说明
SseKmsKey	string	可选	<p>参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p>约束限制: 有效值支持两种格式: 1. regionID:domainID(账号ID):key/key_id 2. key_id 其中:</p> <ul style="list-style-type: none">• regionID是使用密钥所属region的ID, 可在地区和终端节点页面获取;• domainID是使用密钥所属账号的账号ID, 获取方法参见如何获取账号ID和用户ID? (Node.js SDK);• key_id是从数据加密服务创建的密钥ID, 获取方法请参见查看密钥。 <p>取值范围: 无</p> <p>默认取值:</p> <ul style="list-style-type: none">• 如果用户没有提供该头域, 那么默认的主密钥将会被使用。• 如果默认主密钥不存在, 将默认创建并使用。
SseC	string	当使用SSE-C方式时, 必选	<p>参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p> <p>约束限制: 仅支持AES256。</p> <p>取值范围: 可选值: AES256, 即选择SSE-C方式加密对象。</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
SseCKey	string	作为请求参数时必选	<p>参数解释: SSE-C方式下加密使用的密钥, 该密钥用于加密对象。</p> <p>约束限制: 该头域由256bit的密钥经过Base64编码得到, 示例: K7QkYpBkM5+hca27fsNkUnNVao bncnLht/rCB2o/9Cw=</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
Metadata	object	可选	<p>参数解释: 待上传对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据, 以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时, 加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为: 每个键和值的UTF-8 编码中的字节总数。 自定义元数据的key值不区分大小写, OBS统一转为小写进行存储。value值区分大小写。 自定义元数据key-value对都必须符合US-ASCII。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。 <p>取值范围: 无</p> <p>默认取值: 无</p>

表 6-49 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限，其他任何人都没有访问权限。
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	桶公共读，桶内对象公共读。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadDelivered不能应用于对象。
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	桶公共读写，桶内对象公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadWriteDelivered不能应用于对象。

常量名	原始值	说明
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	<p>设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。</p> <p>默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。</p>

返回结果

表 6-50 返回结果

参数类型	描述
<p>表6-51</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表6-51。</p>

表 6-51 Response

参数名称	参数类型	描述
CommonMsg	ICCommonMsg	<p>参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICCommonMsg。</p>
InterfaceResult	表6-53	<p>参数解释： 操作成功后的结果数据，详见表6-53。</p> <p>约束限制： 当Status大于300时为空。</p>

表 6-52 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 6-53 UploadFileOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
ETag	string	参数解释: 合并段后根据各个段的ETag值计算出的结果。对象的base64编码的128位MD5摘要。ETag是对象内容的唯一标识，可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A，下载对象时ETag为B，则说明对象内容发生了变化。ETag只反映变化的内容，而不是其元数据。上传的对象或复制操作创建的对象，都有唯一的ETag。 约束限制: 当对象是服务端加密的对象时，ETag值不是对象的MD5值。

参数名称	参数类型	描述
Bucket	string	参数解释: 合并段所在的桶名。
Key	string	参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。
Location	string	参数解释: 合并段后得到的对象的url。 例如: https://example-Bucket.obs.regions.myhuaweicloud.com/example-Object
VersionId	string	参数解释: 合并段后得到的对象版本号。如果桶的多版本状态为开启，则会返回对象的版本号。
SseKms	string	参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。
SseKmsKey	string	参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。
SseC	string	参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。
SseCKeyMd5	string	参数解释: SSE-C方式下加密使用密钥的MD5值，该值用于验证密钥传输过程中是否出错。 约束限制: 由密钥值经过MD5加密再经过Base64编码后得到，示例: 4XvB3tbNTN+tIEVa0/fGaQ==

代码示例

本示例用于断点上传example/objectname对象到examplebucket桶中。

```
// 引入obs库
// 使用npm安装
```

```
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function uploadFile() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: 'examplebucket',
      // 指定上传对象, 此处以 example/objectname 为例
      Key: 'example/objectname',
      // 指定待上传的本地文件, 此处以/tmp/objectname为例
      UploadFile: 'localfile',
      // 指定是否开启断点续传模式, 此处以true为例。默认为false, 表示不开启
      EnableCheckpoint: true,
      // 指定分段大小, 单位字节。此处以每段9M为例
      PartSize: 9 * 1024 * 1024,
      // 指定分段下载时的最大并发数, 此处以并发数5为例
      TaskNum: 5
    };
    // 断点续传上传对象
    const result = await obsClient.uploadFile(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Upload file(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      fmt.Printf("ETag:%s\n", result.InterfaceResult.ETag)
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}
uploadFile();
```

6.10 基于表单上传(Node.js SDK)

须知

开发过程中, 您有任何问题可以在github上[提交issue](#)。

功能说明

基于表单上传是使用HTML表单形式上传对象到指定桶中，对象最大不能超过5GB。

您可以通过ObsClient.createPostSignatureSync生成基于表单上传的请求参数。使用Node.js代码模拟表单上传的完整代码示例，参见[post-object-sample](#)。您可以通过如下步骤进行表单上传：

步骤1 使用ObsClient.createPostSignatureSync生成用于鉴权的请求参数。

步骤2 准备表单HTML页面。

步骤3 将生成的请求参数填入HTML页面。

步骤4 选择本地文件，进行表单上传。

----结束

方法定义

```
ObsClient.createPostSignatureSync(params)
```

请求参数

表 6-54 请求参数列表

参数名称	类型	是否必选	说明
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	类型	是否必选	说明
Key	string	必选	参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。 约束限制: 无 取值范围: 长度大于0且不超过1024的字符串。 默认取值: 无
Expires	Number	可选	参数解释: 表单上传鉴权的过期时间。 约束限制: 无 取值范围: 正整数。单位：秒。 默认取值: 300
FormParams	Object	可选	参数解释: 除key、policy、signature外，表单上传时的其他参数。 约束限制: 无 取值范围: <ul style="list-style-type: none">• acl• cache-control• content-type• content-disposition• content-encoding• expires 默认取值: 无

返回结果

表 6-55 返回结果

参数名称	参数类型	描述
OriginPolicy	String	参数解释: Policy未经过base64之前的值, 仅用于校验。
Policy	String	参数解释: 表单中的policy。
Signature	String	参数解释: 表单中的signature。

代码示例

以下代码展示了如何生成基于表单上传的请求参数:

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

function createSignedUrlSync() {
  const params = {
    // 指定存储桶名称
    Bucket: "examplebucket",
    // 指定对象, 此处以 example/objectname 为例
    Key: "example/objectname",
    // 指定签名URL的过期时间, 这里以3600为例, 单位为秒
    Expires: 3600,
    // 指定请求中携带的头域
    FormParams: {
      'x-obs-acl': 'public-read',
      'content-type': 'text/plain'
    }
  };
  // 生成上传对象的带授权信息的URL
  const res = obsClient.createPostSignatureSync(params);
  console.log("OriginPolicy : %s", res.OriginPolicy);
  console.log("Policy: %s", res.Policy);
  console.log("Signature: %v", res.Signature);
};

createSignedUrlSync()
```


示例表单HTML代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>

<form action="http://bucketname.your-endpoint/" method="post" enctype="multipart/form-data">
Object key
<!-- 对象名 -->
<input type="text" name="key" value="objectname" />
<p>
ACL
<!-- 对象ACL权限 -->
<input type="text" name="x-obs-acl" value="public-read" />
<p>
Content-Type
<!-- 对象MIME类型 -->
<input type="text" name="content-type" value="text/plain" />
<p>
<!-- policy的base64编码值 -->
<input type="hidden" name="policy" value="**** Provide your policy ****" />
<!-- AK -->
<input type="hidden" name="AccessKeyId" value="**** Provide your access key ****"/>
<!-- 签名串信息 -->
<input type="hidden" name="signature" value="**** Provide your signature ****"/>

<input name="file" type="file" />
<input name="submit" value="Upload" type="submit" />
</form>
</body>
</html>
```

说明

您可以直接下载表单HTML示例[PostDemo](#)。

7 下载对象(Node.js SDK)

7.1 对象下载简介(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

OBS Node.js SDK提供了丰富的对象下载接口，可以通过以下方式下载对象：

- [文本下载\(Node.js SDK\)](#)
- [流式下载\(Node.js SDK\)](#)
- [文件下载\(Node.js SDK\)](#)
- [范围下载\(Node.js SDK\)](#)
- [断点续传下载\(Node.js SDK\)](#)

您可以通过ObsClient.getObject下载对象。

7.2 文本下载(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

您可以根据需要将存储在OBS中的对象以文本形式下载到本地。

接口约束

- 您必须是桶拥有者或拥有下载对象的权限，才能下载对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:GetObject权限，如果使用桶策略

则需授予GetObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。

- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 对于存储类别为归档存储的对象，需要确认对象的状态为“已恢复”才能对其进行下载。

方法定义

ObsClient.getObject(params)

请求参数

表 7-1 请求参数列表

参数名称	类型	是否必选	说明
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。• 桶命名规则如下：<ul style="list-style-type: none">- 3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。- 禁止使用IP地址。- 禁止以“-”或“.”开头及结尾。- 禁止两个“.”相邻（如：“my..bucket”）。- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。• 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	类型	是否必选	说明
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如：G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
SaveAsFile	string	可选	<p>参数解释: 下载对象的目标路径，包含文件名。</p> <p>约束限制: 包含文件名，如aa/bb.txt</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
SaveAsStream	boolean	可选	参数解释: 是否将对象以流 (stream.Readable) 的形式返回。 取值范围: <ul style="list-style-type: none">• true: 将对象以流 (stream.Readable) 的形式返回。• false: 不将对象以流 (stream.Readable) 的形式返回。 默认取值: false
ProgressCallback	function	可选	参数解释: 获取下载进度的回调函数。 说明 该回调函数依次包含三个参数: 已下载的字节数、总字节数、已使用的时间 (单位: 秒)。 约束限制: 无 取值范围: 无 默认取值: 无
IfMatch	string	可选	参数解释: 指定一个预设的Etag值, 如果下载对象的Etag值与该参数值相同, 则返回对象内容, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无

参数名称	类型	是否必选	说明
IfModifiedSince	string	可选	<p>参数解释: 如果对象的修改时间晚于该参数值指定的时间, 则返回对象内容, 否则返回异常码。</p> <p>约束限制: 该参数值必须符合http://www.ietf.org/rfc/rfc2616.txt规定的HTTP时间格式。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
IfNoneMatch	string	可选	<p>参数解释: 指定一个预设的Etag值, 如果下载对象的ETag值与该参数值不相同, 则返回对象内容, 否则返回错误。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
IfUnmodifiedSince	string	可选	<p>参数解释: 如果对象的修改时间早于该参数值指定的时间, 则返回对象内容, 否则返回异常码</p> <p>约束限制: 该参数值必须符合http://www.ietf.org/rfc/rfc2616.txt规定的HTTP时间格式。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
Range	string	可选	<p>参数解释: 指定下载的范围。例如: 0-999, 即对象的第1个字节到1000字节。</p> <p>约束限制: 如果Range的最大长度超出对象长度-1, 仍旧取对象长度-1。</p> <p>取值范围: 取值区间: [0, 对象长度-1], 格式: bytes=x-y。</p> <p>默认取值: 无</p>
Origin	string	可选	<p>参数解释: 预请求指定的跨域请求Origin (通常为域名)。</p> <p>约束限制: 每个匹配规则允许使用最多一个“*”通配符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
RequestHeader	string	可选	<p>参数解释: 跨域请求可以使用的HTTP头域。只有匹配上允许的头域中的配置, 才被视为是合法的CORS请求。</p> <p>约束限制: 每个头域可填写一个“*”通配符, 不支持&、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseCacheControl	string	可选	<p>参数解释: 重写响应中的Cache-Control头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Cache-Control的取值。</p> <p>默认取值: 无</p>
ResponseContentDisposition	string	可选	<p>参数解释: 重写响应中的Content-Disposition头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Disposition的取值。</p> <p>默认取值: 无</p>
ResponseContentEncoding	string	可选	<p>参数解释: 重写响应中的Content-Encoding头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Encoding的取值。</p> <p>默认取值: 无</p>
ResponseContentLanguage	string	可选	<p>参数解释: 重写响应中的Content-Language头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Language的取值。</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseContentType	string	可选	参数解释: 重写响应中的Content-Type头。 约束限制: 无 取值范围: 常见的Content-Type (MIME) 列表 参见 如何理解Content-Type (MIME) (Node.js SDK) 默认取值: 无
ResponseExpires	string	可选	参数解释: 重写响应中的Expires头。 约束限制: 无 取值范围: 参见HTTP标准头域Expires的取值。 默认取值: 无
ImageProcess	string	可选	参数解释: 图片处理参数, 描述针对对象的图片处理命令或处理样式。例如表示对图片依次进行缩放、旋转, 取值: image/resize,m_fixed,w_100,h_100/rotate,90。 约束限制: 无 取值范围: 命令方式: image/命令参数。 样式方式: style/样式名称。 详细参数说明参见 处理图片 。 默认取值: 如果不输入处理命令, 将返回原图。

参数名称	类型	是否必选	说明
SseC	string	当使用SSE-C方式时，必选	参数解释： 以SSE-C方式解密对象。 约束限制： 仅支持AES256。 取值范围： 可选值：AES256。 默认取值： 无
SseCKey	string	当使用SSE-C方式时，必选	参数解释： SSE-C方式下解密的密钥，该密钥用于解密对象。 约束限制： 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= 取值范围： 无 默认取值： 无

📖 说明

- 如果包含IfUnmodifiedSince并且不符合，或者包含IfMatch并且不符合，抛出异常中HTTP状态码为：412 precondition failed。
- 如果包含IfModifiedSince并且不符合，或者包含IfNoneMatch并且不符合，抛出异常中HTTP状态码为：304 Not Modified。

返回结果

表 7-2 返回结果

参数类型	描述
表7-3 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见 表7-3 。

表 7-3 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	表7-5	参数解释: 操作成功后的结果数据, 详见 表7-5 。 约束限制: 当Status大于300时为空。

表 7-4 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 7-5 GetObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
DeleteMarker	string	参数解释: 标识删除的对象是否是删除标记。
LastModified	string	参数解释: 对象的最近一次修改时间。
ContentLength	string	参数解释: 对象数据的长度。
CacheControl	string	参数解释: 响应中的Cache-Control头。
ContentDisposition	string	参数解释: 响应中的Content-Disposition头。
ContentEncoding	string	参数解释: 响应中的Content-Encoding头
ContentLanguage	string	参数解释: 响应中的Content-Language头
ContentType	string	参数解释: 对象的MIME类型。
Expires	string	参数解释: 响应中的Expires头。
ETag	string	参数解释: 对象的ETag值。
VersionId	string	参数解释: 对象的版本号。
WebsiteRedirectLocation	string	参数解释: 当桶设置了Website配置, 该参数指明对象的重定向地址。
StorageClass	string	参数解释: 对象的存储类型, 当对象存储类型是标准存储时, 该值为空。
Restore	string	参数解释: 归档存储类型对象的恢复状态。
Expiration	string	参数解释: 对象的详细过期信息。

参数名称	参数类型	描述
Content	string stream.Readable	参数解释: 对象的内容。当设置了SaveAsFile时该值为空；当设置了SaveAsStream且为true时该值为stream.Readable对象；当SaveAsFile和SaveAsStream都未设置时为Buffer对象。
Metadata	object	参数解释: 对象自定义元数据。需要在桶的CORS配置中增加允许响应中可返回的附加头域。例如，新增x-amz-meta-property1以获取自定义元数据property1。

代码示例

本示例用于文本下载对象，以examplebucket/objectname为例。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');
// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: 'example/objectname',
    };

    // 文本下载对象
    const result = await obsClient.getObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("Object Content: %s", result.InterfaceResult.Content);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
```

```
console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
console.log(error);
};
};
getObject();
```

7.3 流式下载(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

您可以根据需要将存储在OBS中的对象流式下载到本地。

接口约束

- 您必须是桶拥有者或拥有下载对象的权限，才能下载对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:GetObject权限，如果使用桶策略则需授予GetObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 对于存储类别为归档存储的对象，需要确认对象的状态为“已恢复”才能对其进行下载。

方法定义

```
ObsClient.getObject(params)
```

请求参数

表 7-6 请求参数列表

参数名称	类型	是否必选	说明
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个人区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	类型	是否必选	说明
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如：G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
SaveAsFile	string	可选	<p>参数解释: 下载对象的目标路径，包含文件名。</p> <p>约束限制: 包含文件名，如aa/bb.txt</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
SaveAsStream	boolean	可选	<p>参数解释: 是否将对象以流 (stream.Readable) 的形式返回。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • true: 将对象以流 (stream.Readable) 的形式返回。 • false: 不将对象以流 (stream.Readable) 的形式返回。 <p>默认取值: false</p>
ProgressCallback	function	可选	<p>参数解释: 获取下载进度的回调函数。</p> <p>说明 该回调函数依次包含三个参数: 已下载的字节数、总字节数、已使用的时间 (单位: 秒)。</p> <p>约束限制: 无</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
IfMatch	string	可选	<p>参数解释: 指定一个预设的Etag值, 如果下载对象的Etag值与该参数值相同, 则返回对象内容, 否则返回错误。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
IfModifiedSince	string	可选	参数解释: 如果对象的修改时间晚于该参数值指定的时间, 则返回对象内容, 否则返回异常码。 约束限制: 该参数值必须符合 http://www.ietf.org/rfc/rfc2616.txt 规定的HTTP时间格式。 取值范围: 无 默认取值: 无
IfNoneMatch	string	可选	参数解释: 指定一个预设的Etag值, 如果下载对象的ETag值与该参数值不相同, 则返回对象内容, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无
IfUnmodifiedSince	string	可选	参数解释: 如果对象的修改时间早于该参数值指定的时间, 则返回对象内容, 否则返回异常码。 约束限制: 该参数值必须符合 http://www.ietf.org/rfc/rfc2616.txt 规定的HTTP时间格式。 取值范围: 无 默认取值: 无

参数名称	类型	是否必选	说明
Range	string	可选	<p>参数解释: 指定下载的范围。例如: 0-999, 即对象的第1个字节到1000字节。</p> <p>约束限制: 如果Range的最大长度超出对象长度-1, 仍旧取对象长度-1。</p> <p>取值范围: 取值区间: [0, 对象长度-1], 格式: bytes=x-y。</p> <p>默认取值: 无</p>
Origin	string	可选	<p>参数解释: 预请求指定的跨域请求Origin (通常为域名)。</p> <p>约束限制: 每个匹配规则允许使用最多一个“*”通配符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
RequestHeader	string	可选	<p>参数解释: 跨域请求可以使用的HTTP头域。只有匹配上允许的头域中的配置, 才被视为是合法的CORS请求。</p> <p>约束限制: 每个头域可填写一个“*”通配符, 不支持&、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseCacheControl	string	可选	<p>参数解释: 重写响应中的Cache-Control头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Cache-Control的取值。</p> <p>默认取值: 无</p>
ResponseContentDisposition	string	可选	<p>参数解释: 重写响应中的Content-Disposition头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Disposition的取值。</p> <p>默认取值: 无</p>
ResponseContentEncoding	string	可选	<p>参数解释: 重写响应中的Content-Encoding头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Encoding的取值。</p> <p>默认取值: 无</p>
ResponseContentLanguage	string	可选	<p>参数解释: 重写响应中的Content-Language头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Language的取值。</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseContentType	string	可选	参数解释: 重写响应中的Content-Type头。 约束限制: 无 取值范围: 常见的Content-Type (MIME) 列表 参见 如何理解Content-Type (MIME) (Node.js SDK) 默认取值: 无
ResponseExpires	string	可选	参数解释: 重写响应中的Expires头。 约束限制: 无 取值范围: 参见HTTP标准头域Expires的取值。 默认取值: 无
ImageProcess	string	可选	参数解释: 图片处理参数, 描述针对对象的图片处理命令或处理样式。例如表示对图片依次进行缩放、旋转, 取值: image/resize,m_fixed,w_100,h_100/rotate,90。 约束限制: 无 取值范围: 命令方式: image/命令参数。 样式方式: style/样式名称。 详细参数说明参见 处理图片 。 默认取值: 如果不输入处理命令, 将返回原图。

参数名称	类型	是否必选	说明
SseC	string	当使用SSE-C方式时，必选	<p>参数解释： 以SSE-C方式解密对象。</p> <p>约束限制： 仅支持AES256。</p> <p>取值范围： 可选值：AES256。</p> <p>默认取值： 无</p>
SseCKey	string	当使用SSE-C方式时，必选	<p>参数解释： SSE-C方式下解密的密钥，该密钥用于解密对象。</p> <p>约束限制： 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw=</p> <p>取值范围： 无</p> <p>默认取值： 无</p>

📖 说明

- 如果包含IfUnmodifiedSince并且不符合或者包含IfMatch并且不符合，抛出异常中HTTP状态码为：412 precondition failed。
- 如果包含IfModifiedSince并且不符合或者包含IfNoneMatch并且不符合，抛出异常中HTTP状态码为：304 Not Modified。

返回结果

表 7-7 返回结果

参数类型	描述
<p>表7-8</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表7-8。</p>

表 7-8 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表7-10	参数解释: 操作成功后的结果数据，详见 表7-10 。 约束限制: 当Status大于300时为空。

表 7-9 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 7-10 GetObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
DeleteMarker	string	参数解释: 标识删除的对象是否是删除标记。
LastModified	string	参数解释: 对象的最近一次修改时间。
ContentLength	string	参数解释: 对象数据的长度。
CacheControl	string	参数解释: 响应中的Cache-Control头。
ContentDisposition	string	参数解释: 响应中的Content-Disposition头。
ContentEncoding	string	参数解释: 响应中的Content-Encoding头
ContentLanguage	string	参数解释: 响应中的Content-Language头
ContentType	string	参数解释: 对象的MIME类型。
Expires	string	参数解释: 响应中的Expires头。
ETag	string	参数解释: 对象的ETag值。
VersionId	string	参数解释: 对象的版本号。
WebsiteRedirectLocation	string	参数解释: 当桶设置了Website配置, 该参数指明对象的重定向地址。
StorageClass	string	参数解释: 对象的存储类型, 当对象存储类型是标准存储时, 该值为空。
Restore	string	参数解释: 归档存储类别对象的恢复状态。
Expiration	string	参数解释: 对象的详细过期信息。

参数名称	参数类型	描述
Content	string stream.Re adable	参数解释: 对象的内容。当设置了SaveAsFile时该值为空；当设置了SaveAsStream且为true时该值为stream.Readable对象；当SaveAsFile和SaveAsStream都未设置时为Buffer对象。
Metadata	object	参数解释: 对象自定义元数据。需要在桶的CORS配置中增加允许响应中可返回的附加头域。例如，新增x-amz-meta-property1以获取自定义元数据property1。

代码示例

本示例用于流式下载对象，以examplebucket/objectname为例。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AK/SK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: 'example/objectname',
      // 指定是对象以流（stream.Readable）的形式返回
      SaveAsStream: true
    };

    // 流式下载对象
    const result = await obsClient.getObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("Object Content:");
      result.InterfaceResult.Content.on('data', (data) => {
        console.log(data.toString());
      });
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
```

```
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};
};

getObject();
```

7.4 文件下载(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

您可以根据需要将存储在OBS中的对象以文件形式下载到本地。

接口约束

- 您必须是桶拥有者或拥有下载对象的权限，才能下载对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:GetObject权限，如果使用桶策略则需授予GetObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 对于存储类别为归档存储的对象，需要确认对象的状态为“已恢复”才能对其进行下载。

方法定义

```
ObsClient.getObject(params)
```

请求参数

表 7-11 请求参数列表

参数名称	类型	是否必选	说明
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	类型	是否必选	说明
Key	string	必选	参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。 约束限制: 无 取值范围: 长度大于0且不超过1024的字符串。 默认取值: 无
VersionId	string	可选	参数解释: 对象的版本号。例如： G001117FCE89978B0000401205D5DC9A。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无
SaveAsFile	string	可选	参数解释: 下载对象的目标路径，包含文件名。 约束限制: 包含文件名，如aa/bb.txt 取值范围: 无 默认取值: 无

参数名称	类型	是否必选	说明
SaveAsStream	boolean	可选	参数解释: 是否将对象以流 (stream.Readable) 的形式返回。 取值范围: <ul style="list-style-type: none">• true: 将对象以流 (stream.Readable) 的形式返回。• false: 不将对象以流 (stream.Readable) 的形式返回。 默认取值: false
ProgressCallback	function	可选	参数解释: 获取下载进度的回调函数。 说明 该回调函数依次包含三个参数: 已下载的字节数、总字节数、已使用的时间 (单位: 秒)。 约束限制: 无 取值范围: 无 默认取值: 无
IfMatch	string	可选	参数解释: 指定一个预设的Etag值, 如果下载对象的Etag值与该参数值相同, 则返回对象内容, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无

参数名称	类型	是否必选	说明
IfModifiedSince	string	可选	参数解释: 如果对象的修改时间晚于该参数值指定的时间, 则返回对象内容, 否则返回异常码。 约束限制: 该参数值必须符合 http://www.ietf.org/rfc/rfc2616.txt 规定的HTTP时间格式。 取值范围: 无 默认取值: 无
IfNoneMatch	string	可选	参数解释: 指定一个预设的Etag值, 如果下载对象的ETag值与该参数值不相同, 则返回对象内容, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无
IfUnmodifiedSince	string	可选	参数解释: 如果对象的修改时间早于该参数值指定的时间, 则返回对象内容, 否则返回异常码。 约束限制: 该参数值必须符合 http://www.ietf.org/rfc/rfc2616.txt 规定的HTTP时间格式。 取值范围: 无 默认取值: 无

参数名称	类型	是否必选	说明
Range	string	可选	<p>参数解释: 指定下载的范围。例如: 0-999, 即对象的第1个字节到1000字节。</p> <p>约束限制: 如果Range的最大长度超出对象长度-1, 仍旧取对象长度-1。</p> <p>取值范围: 取值区间: [0, 对象长度-1], 格式: bytes=x-y。</p> <p>默认取值: 无</p>
Origin	string	可选	<p>参数解释: 预请求指定的跨域请求Origin (通常为域名)。</p> <p>约束限制: 每个匹配规则允许使用最多一个“*”通配符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
RequestHeader	string	可选	<p>参数解释: 跨域请求可以使用的HTTP头域。只有匹配上允许的头域中的配置, 才被视为是合法的CORS请求。</p> <p>约束限制: 每个头域可填写一个“*”通配符, 不支持&、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseCacheControl	string	可选	<p>参数解释: 重写响应中的Cache-Control头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Cache-Control的取值。</p> <p>默认取值: 无</p>
ResponseContentDisposition	string	可选	<p>参数解释: 重写响应中的Content-Disposition头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Disposition的取值。</p> <p>默认取值: 无</p>
ResponseContentEncoding	string	可选	<p>参数解释: 重写响应中的Content-Encoding头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Encoding的取值。</p> <p>默认取值: 无</p>
ResponseContentLanguage	string	可选	<p>参数解释: 重写响应中的Content-Language头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Language的取值。</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseContentType	string	可选	参数解释: 重写响应中的Content-Type头。 约束限制: 无 取值范围: 常见的Content-Type (MIME) 列表 参见 如何理解Content-Type (MIME) (Node.js SDK) 默认取值: 无
ResponseExpires	string	可选	参数解释: 重写响应中的Expires头。 约束限制: 无 取值范围: 参见HTTP标准头域Expires的取值。 默认取值: 无
ImageProcess	string	可选	参数解释: 图片处理参数, 描述针对对象的图片处理命令或处理样式。例如表示对图片依次进行缩放、旋转, 取值: image/resize,m_fixed,w_100,h_100/rotate,90。 约束限制: 无 取值范围: 命令方式: image/命令参数。 样式方式: style/样式名称。 详细参数说明参见 处理图片 。 默认取值: 如果不输入处理命令, 将返回原图。

参数名称	类型	是否必选	说明
SseC	string	当使用SSE-C方式时，必选	参数解释： 以SSE-C方式解密对象。 约束限制： 仅支持AES256。 取值范围： 可选值：AES256。 默认取值： 无
SseCKey	string	当使用SSE-C方式时，必选	参数解释： SSE-C方式下解密的密钥，该密钥用于解密对象。 约束限制： 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= 取值范围： 无 默认取值： 无

📖 说明

- 如果包含IfUnmodifiedSince并且不符合或者包含IfMatch并且不符合，抛出异常中HTTP状态码为：412 precondition failed。
- 如果包含IfModifiedSince并且不符合或者包含IfNoneMatch并且不符合，抛出异常中HTTP状态码为：304 Not Modified。

返回结果

表 7-12 返回结果

参数类型	描述
表7-13 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见 表7-13 。

表 7-13 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表7-15	参数解释: 操作成功后的结果数据，详见 表7-15 。 约束限制: 当Status大于300时为空。

表 7-14 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 7-15 GetObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
DeleteMarker	string	参数解释: 标识删除的对象是否是删除标记。
LastModified	string	参数解释: 对象的最近一次修改时间。
ContentLength	string	参数解释: 对象数据的长度。
CacheControl	string	参数解释: 响应中的Cache-Control头。
ContentDisposition	string	参数解释: 响应中的Content-Disposition头。
ContentEncoding	string	参数解释: 响应中的Content-Encoding头
ContentLanguage	string	参数解释: 响应中的Content-Language头
ContentType	string	参数解释: 对象的MIME类型。
Expires	string	参数解释: 响应中的Expires头。
ETag	string	参数解释: 对象的ETag值。
VersionId	string	参数解释: 对象的版本号。
WebsiteRedirectLocation	string	参数解释: 当桶设置了Website配置, 该参数指明对象的重定向地址。
StorageClass	string	参数解释: 对象的存储类型, 当对象存储类型是标准存储时, 该值为空。
Restore	string	参数解释: 归档存储类型对象的恢复状态。
Expiration	string	参数解释: 对象的详细过期信息。

参数名称	参数类型	描述
Content	string stream.Re adable	参数解释: 对象的内容。当设置了SaveAsFile时该值为空；当设置了SaveAsStream且为true时该值为stream.Readable对象；当SaveAsFile和SaveAsStream都未设置时为Buffer对象。
Metadata	object	参数解释: 对象自定义元数据。需要在桶的CORS配置中增加允许响应中可返回的附加头域。例如，新增x-amz-meta-property1以获取自定义元数据property1。

代码示例

本示例用于文件下载对象，以examplebucket/objectname为例。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: 'example/objectname',
      // 指定下载对象的目标路径
      SaveAsFile: 'localfile'
    };

    // 文件下载对象
    const result = await obsClient.getObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
  }
}
```

```
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};
getObject();
```

7.5 范围下载(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

您可以根据需要将存储在OBS中的对象下载到本地。如果只需要下载对象的其中一部分数据，可以使用范围下载，下载指定范围的数据。如果指定的下载范围是0~1000，则返回第0到第1000个字节的数据，包括第1000个，共1001字节的数据，即[0, 1000]。如果指定的范围无效，则返回整个对象的数据。可以利用范围下载并发下载大对象，详细代码示例请参考[concurrent-download-object-sample](#)。

接口约束

- 您必须是桶拥有者或拥有下载对象的权限，才能下载对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:GetObject权限，如果使用桶策略则需授予GetObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 对于存储类别为归档存储的对象，需要确认对象的状态为“已恢复”才能对其进行下载。
- 如果指定的范围无效（比如开始位置、结束位置为负数，大于文件大小），则会返回整个对象。

方法定义

```
ObsClient.getObject(params)
```

请求参数

表 7-16 请求参数列表

参数名称	类型	是否必选	说明
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	类型	是否必选	说明
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如： G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
SaveAsFile	string	可选	<p>参数解释: 下载对象的目标路径，包含文件名。</p> <p>约束限制: 包含文件名，如aa/bb.txt</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
SaveAsStream	boolean	可选	参数解释: 是否将对象以流 (stream.Readable) 的形式返回。 取值范围: <ul style="list-style-type: none">• true: 将对象以流 (stream.Readable) 的形式返回。• false: 不将对象以流 (stream.Readable) 的形式返回。 默认取值: false
ProgressCallback	function	可选	参数解释: 获取下载进度的回调函数。 说明 该回调函数依次包含三个参数: 已下载的字节数、总字节数、已使用的时间 (单位: 秒)。 约束限制: 无 取值范围: 无 默认取值: 无
IfMatch	string	可选	参数解释: 指定一个预设的Etag值, 如果下载对象的Etag值与该参数值相同, 则返回对象内容, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无

参数名称	类型	是否必选	说明
IfModifiedSince	string	可选	参数解释: 如果对象的修改时间晚于该参数值指定的时间,则返回对象内容,否则返回异常码。 约束限制: 该参数值必须符合 http://www.ietf.org/rfc/rfc2616.txt 规定的HTTP时间格式。 取值范围: 无 默认取值: 无
IfNoneMatch	string	可选	参数解释: 指定一个预设的Etag值,如果下载对象的ETag值与该参数值不相同,则返回对象内容,否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无
IfUnmodifiedSince	string	可选	参数解释: 如果对象的修改时间早于该参数值指定的时间,则返回对象内容,否则返回异常码。 约束限制: 该参数值必须符合 http://www.ietf.org/rfc/rfc2616.txt 规定的HTTP时间格式。 取值范围: 无 默认取值: 无

参数名称	类型	是否必选	说明
Range	string	可选	<p>参数解释: 指定下载的范围。例如: 0-999, 即对象的第1个字节到1000字节。</p> <p>约束限制: 如果Range的最大长度超出对象长度-1, 仍旧取对象长度-1。</p> <p>取值范围: 取值区间: [0, 对象长度-1], 格式: bytes=x-y, 例如: bytes=1-1000。</p> <p>默认取值: 无</p>
Origin	string	可选	<p>参数解释: 预请求指定的跨域请求Origin (通常为域名)。</p> <p>约束限制: 每个匹配规则允许使用最多一个“*”通配符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
RequestHeader	string	可选	<p>参数解释: 跨域请求可以使用的HTTP头域。只有匹配上允许的头域中的配置, 才被视为是合法的CORS请求。</p> <p>约束限制: 每个头域可填写一个“*”通配符, 不支持&、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseCacheControl	string	可选	<p>参数解释: 重写响应中的Cache-Control头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Cache-Control的取值。</p> <p>默认取值: 无</p>
ResponseContentDisposition	string	可选	<p>参数解释: 重写响应中的Content-Disposition头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Disposition的取值。</p> <p>默认取值: 无</p>
ResponseContentEncoding	string	可选	<p>参数解释: 重写响应中的Content-Encoding头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Encoding的取值。</p> <p>默认取值: 无</p>
ResponseContentLanguage	string	可选	<p>参数解释: 重写响应中的Content-Language头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Language的取值。</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseContentType	string	可选	<p>参数解释: 重写响应中的Content-Type头。</p> <p>约束限制: 无</p> <p>取值范围: 常见的Content-Type (MIME) 列表 参见如何理解Content-Type (MIME) (Node.js SDK)</p> <p>默认取值: 无</p>
ResponseExpires	string	可选	<p>参数解释: 重写响应中的Expires头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Expires的取值。</p> <p>默认取值: 无</p>
ImageProcess	string	可选	<p>参数解释: 图片处理参数, 描述针对对象的图片处理命令或处理样式。例如表示对图片依次进行缩放、旋转, 取值: image/resize,m_fixed,w_100,h_100/rotate,90。</p> <p>约束限制: 无</p> <p>取值范围: 命令方式: image/命令参数。 样式方式: style/样式名称。 详细参数说明参见处理图片。</p> <p>默认取值: 如果不输入处理命令, 将返回原图。</p>

参数名称	类型	是否必选	说明
SseC	string	当使用SSE-C方式时，必选	<p>参数解释： 以SSE-C方式解密对象。</p> <p>约束限制： 仅支持AES256。</p> <p>取值范围： 可选值：AES256。</p> <p>默认取值： 无</p>
SseCKey	string	当使用SSE-C方式时，必选	<p>参数解释： SSE-C方式下解密的密钥，该密钥用于解密对象。</p> <p>约束限制： 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw=</p> <p>取值范围： 无</p> <p>默认取值： 无</p>

📖 说明

- 如果包含IfUnmodifiedSince并且不符合或者包含IfMatch并且不符合，抛出异常中HTTP状态码为：412 precondition failed。
- 如果包含IfModifiedSince并且不符合或者包含IfNoneMatch并且不符合，抛出异常中HTTP状态码为：304 Not Modified。

返回结果

表 7-17 返回结果

参数类型	描述
<p>表7-18</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表7-18。</p>

表 7-18 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	表7-20	参数解释: 操作成功后的结果数据, 详见 表7-20 。 约束限制: 当Status大于300时为空。

表 7-19 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 7-20 GetObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
DeleteMarker	string	参数解释: 标识删除的对象是否是删除标记。
LastModified	string	参数解释: 对象的最近一次修改时间。
ContentLength	string	参数解释: 对象数据的长度。
CacheControl	string	参数解释: 响应中的Cache-Control头。
ContentDisposition	string	参数解释: 响应中的Content-Disposition头。
ContentEncoding	string	参数解释: 响应中的Content-Encoding头
ContentLanguage	string	参数解释: 响应中的Content-Language头
ContentType	string	参数解释: 对象的MIME类型。
Expires	string	参数解释: 响应中的Expires头。
ETag	string	参数解释: 对象的ETag值。
VersionId	string	参数解释: 对象的版本号。
WebsiteRedirectLocation	string	参数解释: 当桶设置了Website配置, 该参数指明对象的重定向地址。
StorageClass	string	参数解释: 对象的存储类型, 当对象存储类型是标准存储时, 该值为空。
Restore	string	参数解释: 归档存储类型对象的恢复状态。
Expiration	string	参数解释: 对象的详细过期信息。

参数名称	参数类型	描述
Content	string stream.Readable	参数解释: 对象的内容。当设置了SaveAsFile时该值为空；当设置了SaveAsStream且为true时该值为stream.Readable对象；当SaveAsFile和SaveAsStream都未设置时为Buffer对象。
Metadata	object	参数解释: 对象自定义元数据。需要在桶的CORS配置中增加允许响应中可返回的附加头域。例如，新增x-amz-meta-property1以获取自定义元数据property1。

代码示例

本示例用于范围下载对象，以examplebucket/objectname为例并且指定下载范围是0~1000。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: 'example/objectname',
      // 指定下载范围
      Range: 'bytes=0-1000'
    };
    // 范围下载对象
    const result = await obsClient.getObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log('Object Content: %s', result.InterfaceResult.Content);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
  }
}
```

```
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};
getObject();
```

7.6 限定条件下下载(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

您可以根据需要将存储在OBS中的对象在满足限定条件的前提下下载到本地。

接口约束

- 您必须是桶拥有者或拥有下载对象的权限，才能下载对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:GetObject权限，如果使用桶策略则需授予GetObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 对于存储类别为归档存储的对象，需要确认对象的状态为“已恢复”才能对其进行下载。

方法定义

```
ObsClient.getObject(params)
```

请求参数

表 7-21 请求参数列表

参数名称	类型	是否必选	说明
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	类型	是否必选	说明
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如：G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
SaveAsFile	string	可选	<p>参数解释: 下载对象的目标路径，包含文件名。</p> <p>约束限制: 包含文件名，如aa/bb.txt</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
SaveAsStream	boolean	可选	参数解释: 是否将对象以流 (stream.Readable) 的形式返回。 取值范围: <ul style="list-style-type: none">• true: 将对象以流 (stream.Readable) 的形式返回。• false: 不将对象以流 (stream.Readable) 的形式返回。 默认取值: false
ProgressCallback	function	可选	参数解释: 获取下载进度的回调函数。 约束限制: 无 取值范围: 无 默认取值: 无 说明 该回调函数依次包含三个参数: 已下载的字节数、总字节数、已使用的时间 (单位: 秒)。
IfMatch	string	可选	参数解释: 指定一个预设的Etag值, 如果下载对象的Etag值与该参数值相同, 则返回对象内容, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无

参数名称	类型	是否必选	说明
IfModifiedSince	string	可选	参数解释: 如果对象的修改时间晚于该参数值指定的时间, 则返回对象内容, 否则返回异常码。 约束限制: 该参数值必须符合http://www.ietf.org/rfc/rfc2616.txt规定的HTTP时间格式。 取值范围: 无 默认取值: 无
IfNoneMatch	string	可选	参数解释: 指定一个预设的Etag值, 如果下载对象的ETag值与该参数值不相同, 则返回对象内容, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无
IfUnmodifiedSince	string	可选	参数解释: 如果对象的修改时间早于该参数值指定的时间, 则返回对象内容, 否则返回异常码 约束限制: 该参数值必须符合http://www.ietf.org/rfc/rfc2616.txt规定的HTTP时间格式。 取值范围: 无 默认取值: 无

参数名称	类型	是否必选	说明
Range	string	可选	<p>参数解释: 指定下载的范围。例如: 0-999, 即对象的第1个字节到1000字节。</p> <p>约束限制: 如果Range的最大长度超出对象长度-1, 仍旧取对象长度-1。</p> <p>取值范围: 取值区间: [0, 对象长度-1], 格式: bytes=x-y。</p> <p>默认取值: 无</p>
Origin	string	可选	<p>参数解释: 预请求指定的跨域请求Origin (通常为域名)。</p> <p>约束限制: 每个匹配规则允许使用最多一个“*”通配符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
RequestHeader	string	可选	<p>参数解释: 跨域请求可以使用的HTTP头域。只有匹配上允许的头域中的配置, 才被视为是合法的CORS请求。</p> <p>约束限制: 每个头域可填写一个“*”通配符, 不支持&、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseCacheControl	string	可选	<p>参数解释: 重写响应中的Cache-Control头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Cache-Control的取值。</p> <p>默认取值: 无</p>
ResponseContentDisposition	string	可选	<p>参数解释: 重写响应中的Content-Disposition头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Disposition的取值。</p> <p>默认取值: 无</p>
ResponseContentEncoding	string	可选	<p>参数解释: 重写响应中的Content-Encoding头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Encoding的取值。</p> <p>默认取值: 无</p>
ResponseContentLanguage	string	可选	<p>参数解释: 重写响应中的Content-Language头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Language的取值。</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseContentType	string	可选	参数解释: 重写响应中的Content-Type头。 约束限制: 无 取值范围: 常见的Content-Type (MIME) 列表 参见 如何理解Content-Type (MIME) (Node.js SDK) 默认取值: 无
ResponseExpires	string	可选	参数解释: 重写响应中的Expires头。 约束限制: 无 取值范围: 参见HTTP标准头域Expires的取值。 默认取值: 无
ImageProcess	string	可选	参数解释: 图片处理参数, 描述针对对象的图片处理命令或处理样式。例如表示对图片依次进行缩放、旋转, 取值: image/resize,m_fixed,w_100,h_100/rotate,90。 约束限制: 无 取值范围: 命令方式: image/命令参数。 样式方式: style/样式名称。 详细参数说明参见 处理图片 。 默认取值: 如果不输入处理命令, 将返回原图。

参数名称	类型	是否必选	说明
SseC	string	当使用SSE-C方式时，必选	参数解释： 以SSE-C方式解密对象。 约束限制： 仅支持AES256。 取值范围： 可选值：AES256。 默认取值： 无
SseCKey	string	当使用SSE-C方式时，必选	参数解释： SSE-C方式下解密的密钥，该密钥用于解密对象。 约束限制： 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= 取值范围： 无 默认取值： 无

📖 说明

- 如果包含IfUnmodifiedSince并且不符合或者包含IfMatch并且不符合，抛出异常中HTTP状态码为：412 precondition failed。
- 如果包含IfModifiedSince并且不符合或者包含IfNoneMatch并且不符合，抛出异常中HTTP状态码为：304 Not Modified。

返回结果

表 7-22 返回结果

参数类型	描述
表7-23 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见 表7-23 。

表 7-23 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表7-25	参数解释: 操作成功后的结果数据，详见 表7-25 。 约束限制: 当Status大于300时为空。

表 7-24 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 7-25 GetObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
DeleteMarker	string	参数解释: 标识删除的对象是否是删除标记。
LastModified	string	参数解释: 对象的最近一次修改时间。
ContentLength	string	参数解释: 对象数据的长度。
CacheControl	string	参数解释: 响应中的Cache-Control头。
ContentDisposition	string	参数解释: 响应中的Content-Disposition头。
ContentEncoding	string	参数解释: 响应中的Content-Encoding头
ContentLanguage	string	参数解释: 响应中的Content-Language头
ContentType	string	参数解释: 对象的MIME类型。
Expires	string	参数解释: 响应中的Expires头。
ETag	string	参数解释: 对象的ETag值。
VersionId	string	参数解释: 对象的版本号。
WebsiteRedirectLocation	string	参数解释: 当桶设置了Website配置, 该参数指明对象的重定向地址。
StorageClass	string	参数解释: 对象的存储类型, 当对象存储类型是标准存储时, 该值为空。
Restore	string	参数解释: 归档存储类型对象的恢复状态。
Expiration	string	参数解释: 对象的详细过期信息。

参数名称	参数类型	描述
Content	string stream.Re adable	参数解释: 对象的内容。当设置了SaveAsFile时该值为空；当设置了SaveAsStream且为true时该值为stream.Readable对象；当SaveAsFile和SaveAsStream都未设置时为Buffer对象。
Metadata	object	参数解释: 对象自定义元数据。需要在桶的CORS配置中增加允许响应中可返回的附加头域。例如，新增x-amz-meta-property1以获取自定义元数据property1。

代码示例

本示例用于限定条件下载对象，以examplebucket/objectname为例，并且指定该文件的修改时间必须在“Wed, 04 Jul 2018 08:54:53 GMT”后才能下载成功。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');
// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: 'example/objectname',
      // 如果下载对象在给定的日期之后存在修改，才会下载成功。否则返回不带消息主题的304响应
      IfModifiedSince: 'Wed, 04 Jul 2018 08:54:53 GMT'
    };
    // 限定条件下载对象
    const result = await obsClient.getObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("Object Content: %s", result.InterfaceResult.Content);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
  }
}
```

```
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
  console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
  console.log(error);  
};  
};  
};  
getObject();
```

7.7 重写响应头(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

下载对象时，可以重写部分HTTP/HTTPS响应头信息。可重写的响应头信息见下表：

参数	作用
ResponseContentType	重写HTTP/HTTPS响应中的Content-Type
ResponseContentLanguage	重写HTTP/HTTPS响应中的Content-Language
ResponseExpires	重写HTTP/HTTPS响应中的Expires
ResponseCacheControl	重写HTTP/HTTPS响应中的Cache-Control
ResponseContentDisposition	重写HTTP/HTTPS响应中的Content-Disposition
ResponseContentEncoding	重写HTTP/HTTPS响应中的Content-Encoding

代码示例

以下代码展示了如何重写响应头：

```
// 引入obs库  
// 使用npm安装  
const ObsClient = require("esdk-obs-nodejs");  
// 使用源码安装  
// var ObsClient = require('./lib/obs');  
// 创建ObsClient实例  
const obsClient = new ObsClient({  
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html  
  access_key_id: process.env.ACCESS_KEY_ID,  
  secret_access_key: process.env.SECRET_ACCESS_KEY,  
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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  
  // security_token: process.env.SECURITY_TOKEN,  
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写  
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
async function getObject() {  
  try {
```

```
const params = {
  // 指定存储桶名称
  Bucket: "examplebucket",
  // 指定下载对象，此处以 example/objectname 为例
  Key: 'example/objectname',
  // 重写响应头，此处重写Content-Type为例
  ResponseContentType : 'image/jpeg'
};

// 重写响应头
const result = await obsClient.getObject(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  // 获取重写的响应头
  console.log("ContentType:%s", result.InterfaceResult.ContentType);
  return;
}
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};

getObject();
```

7.8 下载归档存储对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

⚠ 注意

重复恢复归档存储数据时在延长恢复有效期的同时，也将会对恢复时产生的恢复费用进行重复收取。产生的标准存储类别的对象副本有效期将会延长，并且收取延长时段产生的标准存储副本费用。

功能说明

如果要获取归档存储对象的内容，需要先将对象恢复，然后再执行下载数据的操作。对象恢复后，会产生一个标准存储类别的对象副本，也就是说会同时存在标准存储类型的对象副本和归档存储对象，在恢复对象的保存时间到期后标准存储类型的对象副本会自动删除。

该接口可以恢复指定桶中的归档存储对象。

接口约束

- 您必须是桶拥有者或拥有恢复归档存储对象的权限，才能恢复归档存储对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:RestoreObject权限，如果使用桶策略则需授予RestoreObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 重复恢复归档存储对象时在延长恢复有效期的同时，也将会对恢复时产生的恢复费用进行重复收取。产生的标准存储类别的对象副本有效期将会延长，并且收取延长时间段产生的标准存储副本费用。

方法定义

```
ObsClient.restoreObject(params)
```

请求参数

表 7-26 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none"> • 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 • 桶命名规则如下： <ul style="list-style-type: none"> - 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 - 禁止使用IP地址。 - 禁止以“-”或“.”开头及结尾。 - 禁止两个“.”相邻（如：“my.bucket”）。 - 禁止“.”和“-”相邻（如：“my-.bucket”和“my-.bucket”）。 • 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: ObsClient.restoreObject中指定的对象必须是归档存储类别，否则调用该接口会报错。</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
Version Id	string	可选	<p>参数解释: 待恢复归档存储对象的版本号。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无，如果不设置则默认指定最新版本的对象。</p>
Days	number	必选	<p>参数解释: 恢复对象后，会生成一个对象的标准存储副本，此参数指定恢复有效期，即标准存储副本的保存时间。</p> <p>约束限制: 无</p> <p>取值范围: [1, 30]，单位：天。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Tier	RestoreTierType	可选	<p>参数解释: 恢复选项，用户可根据需求选择恢复选项，不同的恢复选项恢复耗时不同。</p> <p>约束限制: 无</p> <p>取值范围: 恢复选项可选值详见RestoreTierType。</p> <p>默认取值: Standard</p>

表 7-27 RestoreTierType

常量名	原始值	说明
ObsClient.enums.RestoreTierExpedited	Expedited	快速恢复，归档存储恢复耗时1~5 min。
ObsClient.enums.RestoreTierStandard	Standard	标准恢复，归档存储恢复耗时3~5 h。

返回结果

表 7-28 返回结果

参数类型	描述
<p>表7-29</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表7-29。</p>

表 7-29 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	<p>参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICommonMsg。</p>

参数名称	参数类型	描述
InterfaceResult	表7-31	参数解释: 操作成功后的结果数据, 详见表7-31。 约束限制: 当Status大于300时为空。

表 7-30 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 7-31 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

您可以通过ObsClient.restoreObject恢复归档存储对象。以下代码展示了如何下载归档存储对象：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');
// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function downloadColdObject() {
  // 指定存储桶名称
  const bucketName = "examplebucket";
  // 指定下载对象，此处以 example/objectname 为例
  const objectKey = "example/objectname";
  try {
    // 恢复归档存储对象
    const restoreObjectOutput = await obsClient.restoreObject({
      Bucket: bucketName,
      Key: objectKey,
      // 指定恢复对象的保存时间，此处以1为例，单位天，取值范围：[1, 30]
      Days: 1,
      // 指定恢复选项，此处以obs.RestoreTierExpedited为例，默认为快速恢复
      Tier: obs.enums.RestoreTierExpedited
    });
    if (restoreObjectOutput.CommonMsg.Status > 300) {
      handleMessage(restoreObjectOutput.CommonMsg);
      return;
    };
    // 等待对象恢复
    await sleep(5 * 60);
    // 下载对象
    const getObjectOutput = await obsClient.getObject({
      Bucket: bucketName,
      Key: objectKey,
    });
    if (getObjectOutput.CommonMsg.Status > 300) {
      handleMessage(getObjectOutput.CommonMsg);
      return;
    };
    console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    console.log('Object Content: %s', result.InterfaceResult.Content);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

function handleMessage(commonMsg) {
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
  console.log("Status: %d", commonMsg.Status);
  console.log("Code: %s", commonMsg.Code);
}
```

```
console.log("Message: %s", commonMsg.Message);  
console.log("RequestId: %s", commonMsg.RequestId);  
};  
  
function sleep() {  
  return new Promise(resolve => {  
    setTimeout(resolve, time * 1000);  
  });  
};  
  
downloadColdObject();
```

7.9 断点续传下载(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

断点续传下载接口是利用[范围下载](#)特性实现的，是对范围下载的封装和加强，当下载大对象到本地文件时，经常出现因网络不稳定或程序崩溃导致下载失败的情况。失败后再次重新下载不仅浪费资源，而且当网络不稳定时仍然有下载失败的风险。断点续传下载接口能有效地解决此类问题引起的下载失败，其原理是将待下载的对象分成若干个分段分别下载，并实时地将每段下载结果统一记录在checkpoint文件中，仅当所有分段都下载成功时返回下载成功的结果，否则在回调函数中返回错误码提醒用户再次调用接口进行重新下载（重新下载时因为有checkpoint文件记录当前的下载进度，避免重新下载所有分段，从而节省资源提高效率）。

断点续传下载接口不仅能在失败重下时节省资源提高效率，还因其对分段进行并发下载的机制能加快下载速度，帮助用户快速完成下载业务；且其对用户透明，用户不用关心checkpoint文件的创建和删除、分段任务的切分、并发下载的实现等内部细节。

接口约束

- 您必须是桶拥有者或拥有下载对象的权限，才能下载对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:GetObject权限，如果使用桶策略则需授予GetObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 对于存储类别为归档存储的对象，需要确认对象的状态为“已恢复”才能对其进行下载。

方法定义

```
ObsClient.downloadFile(params)
```

请求参数

表 7-32 请求参数列表

参数名称	类型	是否必选	说明
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	类型	是否必选	说明
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如：G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
DownloadFile	string	可选	<p>参数解释: 下载对象的本地文件全路径。当该值为空时，默认为当前程序的运行目录。</p> <p>约束限制: 包含文件名，如aa/bb.txt</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
PartSize	number	可选	参数解释: 当前段的长度。 约束限制: 无 取值范围: 100KB~5GB, 单位: 字节。 默认取值: 102400字节
TaskNum	number	可选	参数解释: 分段下载时的最大并发数。 约束限制: 无 取值范围: 1~10000 默认取值: 1, 即不设置则默认串行下载。
ProgressCallback	function	可选	参数解释: 获取下载进度的回调函数。 说明 该回调函数依次包含三个参数: 已下载的字节数、总字节数、已使用的时间(单位: 秒)。 约束限制: 无 取值范围: 无 默认取值: 无

参数名称	类型	是否必选	说明
ResumeCallback	function	可选	<p>参数解释: 获取取消断点续传控制参数的回调函数。</p> <p>约束限制: 无</p> <p>取值范围: 无</p> <p>默认取值: 无</p> <p>说明</p> <ul style="list-style-type: none"> 该回调函数有一个参数：取消断点续传下载任务控制参数； 可以调用取消断点续传下载任务控制参数的cancel方法来暂停断点续传下载任务。
EnableCheckpoint	boolean	可选	<p>参数解释: 是否开启断点续传模式。</p> <p>约束限制: 无</p> <p>取值范围:</p> <ul style="list-style-type: none"> true: 开启断点续传模式。 false: 关闭断点续传模式，此时断点续传下载接口退化成对分段下载的简单封装，不会产生checkpoint文件 <p>默认取值: false</p>
CheckpointFile	string	可选	<p>参数解释: 断点续传过程中，会生成一个进度记录文件，文件中会记录段的下载进度和段的相关信息。checkpointFile参数为该记录文件的文件路径。</p> <p>约束限制: 仅在断点续传模式下有效。</p> <p>取值范围: 无</p> <p>默认取值: 当该值为空时，默认为待下载的本地文件的同级目录。</p>

参数名称	类型	是否必选	说明
IfMatch	string	可选	参数解释: 指定一个预设的Etag值, 如果下载对象的ETag值与该参数值相同, 则返回对象内容, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无
IfModifiedSince	string	可选	参数解释: 如果对象的修改时间晚于该参数值指定的时间, 则返回对象内容, 否则返回异常码。 约束限制: 该参数值必须符合 http://www.ietf.org/rfc/rfc2616.txt 规定的HTTP时间格式。 取值范围: 无 默认取值: 无
IfNoneMatch	string	可选	参数解释: 指定一个预设的Etag值, 如果下载对象的ETag值与该参数值不相同, 则返回对象内容, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无

参数名称	类型	是否必选	说明
IfUnmodifiedSince	string	可选	<p>参数解释: 如果对象的修改时间早于该参数值指定的时间, 则返回对象内容, 否则返回异常码。</p> <p>约束限制: 该参数值必须符合http://www.ietf.org/rfc/rfc2616.txt规定的HTTP时间格式。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
SseC	string	当使用SSE-C方式时, 必选	<p>参数解释: 以SSE-C方式解密对象。</p> <p>约束限制: 仅支持AES256。</p> <p>取值范围: 可选值: AES256。</p> <p>默认取值: 无</p>
SseCKey	string	当使用SSE-C方式时, 必选	<p>参数解释: SSE-C方式下解密的密钥, 该密钥用于解密对象。</p> <p>约束限制: 该头域由256bit的密钥经过Base64编码得到, 示例: K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw=</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

说明

- 如果包含IfUnmodifiedSince并且不符合或者包含IfMatch并且不符合, 抛出异常中HTTP状态码为: 412 precondition failed。
- 如果包含IfModifiedSince并且不符合或者包含IfNoneMatch并且不符合, 抛出异常中HTTP状态码为: 304 Not Modified。

返回结果

表 7-33 返回结果

参数类型	描述
表7-34 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见 表7-34 。

表 7-34 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表7-36	参数解释： 操作成功后的结果数据，详见 表7-36 。 约束限制： 当Status大于300时为空。

表 7-35 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释： OBS服务端返回的HTTP状态码。 取值范围： 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 。
Code	string	参数解释： OBS服务端返回的错误码。
Message	string	参数解释： OBS服务端返回的错误描述。
HostId	string	参数解释： OBS服务端返回的请求服务端ID。
RequestId	string	参数解释： OBS服务端返回的请求ID。

Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 7-36 DownloadFileOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
DeleteMarker	string	参数解释: 标识删除的对象是否是删除标记。
LastModified	string	参数解释: 对象的最近一次修改时间。
CacheControl	string	参数解释: 响应中的Cache-Control头。
ContentDisposition	string	参数解释: 响应中的Content-Disposition头。
ContentEncoding	string	参数解释: 响应中的Content-Encoding头
ContentLanguage	string	参数解释: 响应中的Content-Language头
ContentType	string	参数解释: 对象的MIME类型。
Expires	string	参数解释: 响应中的Expires头。
ETag	string	参数解释: 对象的ETag值。
VersionId	string	参数解释: 对象的版本号。
WebsiteRedirectLocation	string	参数解释: 当桶设置了Website配置，该参数指明对象的重定向地址。
StorageClass	string	参数解释: 对象的存储类型，当对象存储类型是标准存储时，该值为空。

参数名称	参数类型	描述
Restore	string	参数解释: 归档存储类型对象的恢复状态。
Expiration	string	参数解释: 对象的详细过期信息。
Metadata	object	参数解释: 对象自定义元数据。需要在桶的CORS配置中增加允许响应中可返回的附加头域。例如, 新增x-amz-meta-property1以获取自定义元数据property1。
SseKms	string	参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。
SseKmsKey	string	参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。
SseC	string	参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。
SseCKeyMd5	string	参数解释: SSE-C方式下加密使用密钥的MD5值, 该值用于验证密钥传输过程中是否出错。 约束限制: 由密钥值经过MD5加密再经过Base64编码后得到, 示例: 4XvB3tbNTN+tIEVa0/fGaQ==

代码示例

本示例用于断点下载examplebucket桶中的example/objectname对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
```

```
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function downloadFile() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: 'examplebucket',
      // 指定对象名, 此处以 example/objectname 为例
      Key: 'example/objectname',
      // 指定下载对象的本地文件全路径, 此处以/tmp/objectname为例。当该值为空时, 默认为当前程序的运行
      // 目录
      DownloadFile: 'localfile',
      // 指定是否开启断点续传模式, 此处以true为例。默认为false, 表示不开启
      EnableCheckpoint: true,
      // 指定分段大小, 单位字节。此处以每段9M为例
      PartSize: 9 * 1024 * 1024,
      // 指定分段下载时的最大并发数, 此处以并发数5为例
      TaskNum: 5
    };
    // 断点续传下载对象
    const result = await obsClient.downloadFile(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Download file(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageClass:%s, ETag:%s, ContentType:%s, ContentLength:%d, LastModified:%s",
        result.InterfaceResult.StorageClass, result.InterfaceResult.ETag, result.InterfaceResult.ContentType,
        result.InterfaceResult.ContentLength, result.InterfaceResult.LastModified,
      );
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
    response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
    attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};
downloadFile();
```

7.10 图片处理(Node.js SDK)

须知

开发过程中, 您有任何问题可以在github上[提交issue](#)。

功能说明

OBS为用户提供了稳定、安全、高效、易用、低成本的图片处理服务。当要下载的对象是图片文件时, 您可以使用**ImageProcess**参数指定图片处理参数对图片文件进行图片剪切、图片缩放、图片水印、格式转换等处理。

接口约束

- 您必须是桶拥有者或拥有下载对象的权限, 才能下载对象。建议使用IAM或桶策略进行授权, 如果使用IAM则需授予obs:object:GetObject权限, 如果使用桶策略

则需授予GetObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。

- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 对于存储类别为归档存储的图片对象，需要确认对象的状态为“已恢复”才能对其进行下载和处理。

方法定义

ObsClient.getObject(params)

请求参数

表 7-37 请求参数列表

参数名称	类型	是否必选	说明
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。• 桶命名规则如下：<ul style="list-style-type: none">- 3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。- 禁止使用IP地址。- 禁止以“-”或“.”开头及结尾。- 禁止两个“.”相邻（如：“my..bucket”）。- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。• 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	类型	是否必选	说明
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如： G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
SaveAsFile	string	可选	<p>参数解释: 下载对象的目标路径，包含文件名。</p> <p>约束限制: 包含文件名，如aa/bb.txt</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
SaveAsStream	boolean	可选	参数解释: 是否将对象以流 (stream.Readable) 的形式返回。 取值范围: <ul style="list-style-type: none">• true: 将对象以流 (stream.Readable) 的形式返回。• false: 不将对象以流 (stream.Readable) 的形式返回。 默认取值: false
ProgressCallback	function	可选	参数解释: 获取下载进度的回调函数。 说明 该回调函数依次包含三个参数: 已下载的字节数、总字节数、已使用的时间 (单位: 秒)。 约束限制: 无 取值范围: 无 默认取值: 无
IfMatch	string	可选	参数解释: 指定一个预设的Etag值, 如果下载对象的Etag值与该参数值相同, 则返回对象内容, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无

参数名称	类型	是否必选	说明
IfModifiedSince	string	可选	参数解释: 如果对象的修改时间晚于该参数值指定的时间, 则返回对象内容, 否则返回异常码。 约束限制: 该参数值必须符合 http://www.ietf.org/rfc/rfc2616.txt 规定的HTTP时间格式。 取值范围: 无 默认取值: 无
IfNoneMatch	string	可选	参数解释: 指定一个预设的Etag值, 如果下载对象的ETag值与该参数值不相同, 则返回对象内容, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无
IfUnmodifiedSince	string	可选	参数解释: 如果对象的修改时间早于该参数值指定的时间, 则返回对象内容, 否则返回异常码 约束限制: 该参数值必须符合 http://www.ietf.org/rfc/rfc2616.txt 规定的HTTP时间格式。 取值范围: 无 默认取值: 无

参数名称	类型	是否必选	说明
Range	string	可选	<p>参数解释: 指定下载的范围。例如: 0-999, 即对象的第1个字节到1000字节。</p> <p>约束限制: 如果Range的最大长度超出对象长度-1, 仍旧取对象长度-1。</p> <p>取值范围: 取值区间: [0, 对象长度-1], 格式: bytes=x-y。</p> <p>默认取值: 无</p>
Origin	string	可选	<p>参数解释: 预请求指定的跨域请求Origin (通常为域名)。</p> <p>约束限制: 每个匹配规则允许使用最多一个“*”通配符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
RequestHeader	string	可选	<p>参数解释: 跨域请求可以使用的HTTP头域。只有匹配上允许的头域中的配置, 才被视为是合法的CORS请求。</p> <p>约束限制: 每个头域可填写一个“*”通配符, 不支持&、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseCacheControl	string	可选	<p>参数解释: 重写响应中的Cache-Control头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Cache-Control的取值。</p> <p>默认取值: 无</p>
ResponseContentDisposition	string	可选	<p>参数解释: 重写响应中的Content-Disposition头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Disposition的取值。</p> <p>默认取值: 无</p>
ResponseContentEncoding	string	可选	<p>参数解释: 重写响应中的Content-Encoding头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Encoding的取值。</p> <p>默认取值: 无</p>
ResponseContentLanguage	string	可选	<p>参数解释: 重写响应中的Content-Language头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Language的取值。</p> <p>默认取值: 无</p>

参数名称	类型	是否必选	说明
ResponseContentType	string	可选	参数解释: 重写响应中的Content-Type头。 约束限制: 无 取值范围: 常见的Content-Type (MIME) 列表 参见 如何理解Content-Type (MIME) (Node.js SDK) 默认取值: 无
ResponseExpires	string	可选	参数解释: 重写响应中的Expires头。 约束限制: 无 取值范围: 参见HTTP标准头域Expires的取值。 默认取值: 无
ImageProcess	string	可选	参数解释: 图片处理参数, 描述针对对象的图片处理命令或处理样式。例如表示对图片依次进行缩放、旋转, 取值: image/resize,m_fixed,w_100,h_100/rotate,90。支持级联处理, 可对图片文件依次实施多条命令 约束限制: 无 取值范围: 命令方式: image/命令参数。 样式方式: style/样式名称。 详细参数说明参见 处理图片 。 默认取值: 如果不输入处理命令, 将返回原图。

参数名称	类型	是否必选	说明
SseC	string	当使用SSE-C方式时，必选	参数解释： 以SSE-C方式解密对象。 约束限制： 仅支持AES256。 取值范围： 可选值：AES256。 默认取值： 无
SseCKey	string	当使用SSE-C方式时，必选	参数解释： SSE-C方式下解密的密钥，该密钥用于解密对象。 约束限制： 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= 取值范围： 无 默认取值： 无

📖 说明

- 如果包含IfUnmodifiedSince并且不符合或者包含IfMatch并且不符合，抛出异常中HTTP状态码为：412 precondition failed。
- 如果包含IfModifiedSince并且不符合或者包含IfNoneMatch并且不符合，抛出异常中HTTP状态码为：304 Not Modified。

返回结果

表 7-38 返回结果

参数类型	描述
表7-39 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见 表7-39 。

表 7-39 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	表7-41	参数解释: 操作成功后的结果数据, 详见 表7-41 。 约束限制: 当Status大于300时为空。

表 7-40 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 7-41 GetObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
DeleteMarker	string	参数解释: 标识删除的对象是否是删除标记。
LastModified	string	参数解释: 对象的最近一次修改时间。
ContentLength	string	参数解释: 对象数据的长度。
CacheControl	string	参数解释: 响应中的Cache-Control头。
ContentDisposition	string	参数解释: 响应中的Content-Disposition头。
ContentEncoding	string	参数解释: 响应中的Content-Encoding头
ContentLanguage	string	参数解释: 响应中的Content-Language头
ContentType	string	参数解释: 对象的MIME类型。
Expires	string	参数解释: 响应中的Expires头。
ETag	string	参数解释: 对象的ETag值。
VersionId	string	参数解释: 对象的版本号。
WebsiteRedirectLocation	string	参数解释: 当桶设置了Website配置, 该参数指明对象的重定向地址。
StorageClass	string	参数解释: 对象的存储类型, 当对象存储类型是标准存储时, 该值为空。
Restore	string	参数解释: 归档存储类型对象的恢复状态。
Expiration	string	参数解释: 对象的详细过期信息。

参数名称	参数类型	描述
Content	string stream.Readable	参数解释: 对象的内容。当设置了SaveAsFile时该值为空；当设置了SaveAsStream且为true时该值为stream.Readable对象；当SaveAsFile和SaveAsStream都未设置时为Buffer对象。
Metadata	object	参数解释: 对象自定义元数据。需要在桶的CORS配置中增加允许响应中可返回的附加头域。例如，新增x-amz-meta-property1以获取自定义元数据property1。

代码示例

本示例用于下载图片对象，以examplebucket/objectname为例，并对图片依次进行缩放、旋转处理。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: 'example/objectname',
      // 对图片依次进行缩放、旋转
      ImageProcess: 'image/resize,m_fixed,w_100,h_100/rotate,90'
    };
    // 图片处理对象
    const result = await obsClient.getObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
  }
}
```

```
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};
getObject();
```

8 管理对象(Node.js SDK)

8.1 设置对象元数据(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

元数据（Metadata）为描述对象属性的信息，是一组名称值对，用作对象管理的一部分。

当前支持系统定义的元数据和自定义元数据。

系统定义的元数据又分为两种类别：系统控制和用户控制。如Last-Modified日期等数据由系统控制，不可修改；如为对象配置的ContentLanguage，用户可以通过接口进行修改。

可以通过本接口添加、修改或删除桶中已经上传的对象的元数据。

接口约束

- 您必须是桶拥有者或拥有设置对象元数据的权限，才能设置对象元数据。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:ModifyObjectMetaData权限，如果使用桶策略则需授予ModifyObjectMetaData权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 当桶开启多版本控制时，最新版本的对象支持设置元数据，历史版本的对象不支持设置元数据。
- 对于存储类别为归档存储的对象，不能对其设置对象的元数据。

方法定义

```
ObsClient.setObjectMetadata(params)
```

请求参数

表 8-1 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p>
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
MetadataDirective	MetadataDirectiveType	必选	<p>参数解释: 复制源对象属性的策略。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的对象属性选项参见 MetadataDirectiveType。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如： G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
CacheControl	string	可选	<p>参数解释: 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Cache-Control的取值。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
ContentDisposition	string	可选	参数解释: 响应中的Content-Disposition头。指定对象被下载时的名称。 约束限制: 无 取值范围: 参见HTTP标准头域Content-Disposition的取值。 默认取值: 无
ContentEncoding	string	可选	参数解释: 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。 约束限制: 无 取值范围: 参见HTTP标准头域Content-Encoding的取值。 默认取值: 无
ContentLanguage	string	可选	参数解释: 响应中的Content-Language头。指定对象被下载时的内容语言格式。 约束限制: 无 取值范围: 参见HTTP标准头域Content-Language的取值。 默认取值: 无

参数名称	参数类型	是否必选	描述
ContentType	string	可选	<p>参数解释: 响应中的Content-Type头。指定对象被下载时的文件类型。</p> <p>约束限制: 无</p> <p>取值范围: 常见的Content-Type (MIME) 列表参见如何理解Content-Type (MIME) (Node.js SDK)</p> <p>默认取值: 无</p>
Expires	string	可选	<p>参数解释: 响应中的Expires头。指定对象被下载时的网页的缓存过期时间。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Expires的取值。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
WebsiteRedirectLocation	string	可选	<p>参数解释: 当桶设置了Website配置,可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。 例如,重定向请求到桶内另一对象: WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL: WebsiteRedirectLocation:http://www.example.com/ OBS将这个值从头域中取出,保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头,长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向,不支持为桶中文件夹下的对象设置重定向。 <p>取值范围: 无</p> <p>默认取值: 无</p>
StorageClass	StorageClassType	可选	<p>参数解释: 指定对象的存储类别。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的存储类别参见StorageClassType。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Metadata	object	可选	<p>参数解释: 对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据，以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时，加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为：每个键和值的UTF-8 编码中的字节总数。 • 自定义元数据的key值不区分大小写，OBS统一转为小写进行存储。value值区分大小写。 • 自定义元数据key-value对都必须符合US-ASCII。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。 <p>取值范围: 无</p> <p>默认取值: 无</p>

表 8-2 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。

常量名	原始值	说明
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

返回结果

表 8-3 返回结果

参数类型	描述
<p>表8-4</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表8-4。</p>

表 8-4 Response

参数名称	参数类型	描述
CommonMsg	ICCommonMsg	<p>参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICCommonMsg。</p>
InterfaceResult	表8-6	<p>参数解释： 操作成功后的结果数据，详见表8-6。</p> <p>约束限制： 当Status大于300时为空。</p>

表 8-5 ICommonMsg

参数名称	参数类型	描述
Status	number	<p>参数解释： OBS服务端返回的HTTP状态码。</p> <p>取值范围： 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。</p>
Code	string	<p>参数解释： OBS服务端返回的错误码。</p>

Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 8-6 SetObjectMetadataOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
MetadataDirective	Metadata Directive Type	参数解释: 复制源对象属性的策略。 取值范围: 可选择对象属性选项参见 MetadataDirectiveType 。
CacheControl	string	参数解释: 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。
ContentDisposition	string	参数解释: 响应中的Content-Disposition头。指定对象被下载时的名称。
ContentEncoding	string	参数解释: 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。
ContentLanguage	string	参数解释: 响应中的Content-Language头。指定对象被下载时的内容语言格式。
ContentType	string	参数解释: 响应中的Content-Type头。指定对象被下载时的文件类型。

参数名称	参数类型	描述
Expires	string	参数解释: 响应中的Expires头。指定对象被下载时的网页的缓存过期时间。
WebsiteRedirectLocation	string	参数解释: 当桶设置了Website配置, 可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。 例如, 重定向请求到桶内另一对象: WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL: WebsiteRedirectLocation:http://www.example.com/ OBS将这个值从头域中取出, 保存在对象的元数据“WebsiteRedirectLocation”中。 约束限制: <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向, 不支持为桶中文件夹下的对象设置重定向。
StorageClass	StorageClassType	参数解释: 复制时设置对象的存储类别。如果未设置此头域, 则以桶的默认存储类别作为对象的存储类别。 取值范围: 可选择的存储类别参见 StorageClassType 。

参数名称	参数类型	描述
Metadata	object	<p>参数解释: 对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据，以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时，加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为：每个键和值的UTF-8 编码中的字节总数。 • 自定义元数据的key值不区分大小写，OBS统一转为小写进行存储。value值区分大小写。 • 自定义元数据key-value对都必须符合US-ASCII。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。

表 8-7 MetadataDirectiveType

常量名	原始值	说明
ObsClient.enums.CopyMetadata	COPY	复制对象时，复制源对象属性。 须知 该值仅在 复制对象(Node.js SDK) 接口中使用
ObsClient.enums.ReplaceMetadata	REPLACE	使用当前请求中携带的头域完整替换，未指定的元数据会被删除。
ObsClient.enums.ReplaceNewMetadata	REPLACE_NEW	对于已经存在值的元数据进行替换，不存在值的元数据进行赋值，未指定的元数据保持不变。 须知 该值仅在 设置对象元数据(Node.js SDK) 接口中使用

表 8-8 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。

常量名	原始值	说明
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

代码示例

本示例用于设置examplebucket桶中的example/objectname对象元数据，并且将对象MIME类型设置成image/jpeg，对象存储类型指定为归档存储，并设置自定义元数据。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AK/SK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setObjectMetadata() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定对象MIME类型，这里以image/jpeg为例
      ContentType: "image/jpeg",
      // 指定对象存储类型，这里以 obsClient.enums.StorageClassCold 为例
      StorageClass: obsClient.enums.StorageClassCold,
      // 指定自定义元数据
      Metadata: { "property1": "property-value1", "property2": "property-value2" }
    };
    // 设置对象元数据
    const result = await obsClient.setObjectMetadata(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set Object(%s)'s metadata successful with bucket(%s)!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
  }
}
```

```
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
  console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
  console.log(error);  
};  
};  
};  
setObjectMetadata();
```

8.2 获取对象元数据(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

元数据（Metadata）为描述对象属性的信息，是一组名称值对，用作对象管理的一部分。

当前仅支持系统定义的元数据。系统定义的元数据又分为两种类别：系统控制和用户控制。如Last-Modified日期等数据由系统控制，不可修改；如为对象配置的ContentLanguage，用户可以通过接口进行修改。

对指定桶中的对象发送HEAD请求，获取对象的元数据信息。

接口约束

- 您必须是桶拥有者或拥有获取对象元数据的权限，才能获取对象元数据。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:GetObject权限，如果使用桶策略则需授予GetObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getObjectMetadata(params)
```


请求参数

表 8-9 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如： G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
Origin	string	可选	<p>参数解释: 预请求指定的跨域请求Origin（通常为域名）。</p> <p>约束限制: 每个匹配规则允许使用最多一个“*”通配符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
RequestHeader	string	可选	<p>参数解释: 跨域请求可以使用的HTTP头域。只有匹配上允许的头域中的配置，才被视为是合法的CORS请求。</p> <p>约束限制: 每个头域可填写一个“*”通配符，不支持&、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
SseC	string	当使用SSE-C方式时，必选	<p>参数解释: 以SSE-C方式解密对象。</p> <p>约束限制: 仅支持AES256。</p> <p>取值范围: 可选值：AES256。</p> <p>默认取值: 无</p>
SseCKey	string	当使用SSE-C方式时，必选	<p>参数解释: SSE-C方式下解密的密钥，该密钥用于解密对象。</p> <p>约束限制: 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw=</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

返回结果

表 8-10 返回结果

参数类型	描述
表8-11 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见表8-11。

表 8-11 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表8-13	参数解释： 操作成功后的结果数据，详见 表8-13 。 约束限制： 当Status大于300时为空。

表 8-12 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释： OBS服务端返回的HTTP状态码。 取值范围： 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 。
Code	string	参数解释： OBS服务端返回的错误码。
Message	string	参数解释： OBS服务端返回的错误描述。
HostId	string	参数解释： OBS服务端返回的请求服务端ID。
RequestId	string	参数解释： OBS服务端返回的请求ID。

Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 8-13 GetObjectMetadataOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
StorageClass	StorageClassType	参数解释: 复制时设置对象的存储类别。如果未设置此头域，则以桶的默认存储类型作为对象的存储类型。 取值范围: 可选择的存储类型参见 StorageClassType 。
AllowOrigin	string	参数解释: 如果请求中的Origin满足桶的CORS规则，则返回CORS规则中的AllowedOrigin。AllowedOrigin指定允许的跨域请求的来源，即允许来自该域名下的请求访问该对象。 约束限制: 表示域名的字符串，每个匹配规则允许使用最多一个“*”通配符。例如： https://*.vbs.example.com 。 默认取值: 无
AllowHeader	string	参数解释: 如果请求的RequestHeader满足桶的CORS规则，则返回CORS规则中的AllowedHeader。AllowedHeader是指允许的跨域请求的头域。只有匹配上允许的头域中的配置，才被视为是合法的CORS请求。 约束限制: 最多可填写一个“*”通配符，不支持&、:、<、空格以及中文字符。 默认取值: 无

参数名称	参数类型	描述
AllowMethod	string	<p>参数解释: 桶CORS规则中的AllowedMethod。 AllowedMethod指允许的跨域请求HTTP方法，即桶和对象的几种操作类型。</p> <p>取值范围: 支持以下HTTP方法:</p> <ul style="list-style-type: none">• GET• PUT• HEAD• POST• DELETE <p>默认取值: 无</p>
ExposeHeader	string	<p>参数解释: 桶CORS规则中的ExposeHeader。 ExposeHeader是指CORS规则允许响应中可返回的附加头域，给客户端提供额外的信息。默认情况下浏览器只能访问以下头域：Content-Length、Content-Type，如果需要访问其他头域，需要在附加头域中配置。</p> <p>约束限制: 不支持*、&、:、<、空格以及中文字符。</p> <p>默认取值: 无</p>
MaxAgeSeconds	number	<p>参数解释: 桶CORS规则中的MaxAgeSeconds。 MaxAgeSeconds指请求来源的客户端可以对跨域请求返回结果的缓存时间。</p> <p>约束限制: 每个CORSRule可以包含至多一个MaxAgeSeconds。</p> <p>取值范围: 0~ (2³¹-1)，单位：秒。</p> <p>默认取值: 100，单位：秒。</p>

参数名称	参数类型	描述
ContentLength	number	参数解释: 对象数据的长度。 取值范围: 0~(2 ⁶³ -1), 单位: 字节。 默认取值: 无
ContentType	string	参数解释: 对象的文件类型(MIME类型)。Content-Type(MIME)用于标识发送或接收数据的类型,浏览器根据该参数来决定数据的打开方式。 取值范围: 常见的Content-Type(MIME)列表参见 如何理解Content-Type(MIME)(Node.js SDK) 默认取值: 如果您在上传Object时未指定Content-Type,SDK会根据指定Object名称的后缀名来判定文件类型并自动填充Content-Type。
LastModified	string	参数解释: 对象的最近一次修改时间。 约束限制: 日期格式必须为ISO8601的格式。例如:2018-01-01T00:00:00.000Z,表示最后一次修改时间为2018-01-01T00:00:00.000Z。 默认取值: 无
ETag	string	参数解释: 对象的ETag值。对象的base64编码的128位MD5摘要。ETag是对象内容的唯一标识,可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A,下载对象时ETag为B,则说明对象内容发生了变化。ETag只反映变化的内容,而不是其元数据。上传的对象或复制操作创建的对象,都有唯一的ETag。 约束限制: 当对象是服务端加密的对象时,ETag值不是对象的MD5值。 取值范围: 长度为32的字符串。 默认取值: 无

参数名称	参数类型	描述
VersionId	string	参数解释: 对象的版本号。 取值范围: 长度为32的字符串。 默认取值: 无
Restore	string	参数解释: 标识对象的恢复状态。对象为归档存储类型，并且处于正在恢复或已经恢复时，会返回此头域。 示例：正在恢复ongoing-request="true"; 已恢复 ongoing-request="false", expiry-date="Wed, 7 Nov 2012 00:00:00 GMT"。其中expiry-date表示对象恢复后的失效时间。 约束限制: 归档存储类型对象的恢复状态，如果对象不为归档存储类型，则该值为空。 默认取值: 无
Expiration	string	参数解释: 对象的详细过期信息。比如: "expiry-date= \"Mon, 11 Sep 2023 00:00:00 GMT\"" 默认取值: 无

参数名称	参数类型	描述
WebsiteRedirectLocation	string	<p>参数解释: 当桶设置了Website配置, 可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。 例如, 重定向请求到桶内另一对象: WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL: WebsiteRedirectLocation:http://www.example.com/ OBS将这个值从头域中取出, 保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向, 不支持为桶中文件夹下的对象设置重定向。 <p>默认取值: 无</p>
Metadata	object	<p>参数解释: 对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据, 以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时, 加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为: 每个键和值的UTF-8 编码中的字节总数。• 自定义元数据的key值不区分大小写, OBS统一转为小写进行存储。value值区分大小写。• 自定义元数据key-value对都必须符合US-ASCII。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。 <p>默认取值: 无</p>

表 8-14 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

代码示例

您可以通过ObsClient.getObjectMetadata来获取对象元数据信息，包括对象长度，对象MIME类型，对象自定义元数据等信息。以下代码展示了如何获取名为examplebucket桶下example/objectname对象的元数据信息：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObjectMetadata() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: 'example/objectname',
    };
    // 获取对象元数据
    const result = await obsClient.getObjectMetadata(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put bucket(%s) successful", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageClass:%s, ETag:%s, ContentType:%s, ContentLength:%d, LastModified:%s\n",
        result.InterfaceResult.StorageClass, result.InterfaceResult.ETag, result.InterfaceResult.ContentType,
        result.InterfaceResult.ContentLength, result.InterfaceResult.LastModified);
    }
  } catch (error) {
    console.error(error);
  }
}
```

```
// 获取对象自定义元数据
console.log("Metadata:%v", result.InterfaceResult.Metadata);
return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
console.log(error);
};
};
};

getObjectMetadata();
```

8.3 设置对象 ACL(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

OBS支持对对象的操作进行权限控制。默认情况下，只有对象的创建者才有该对象的读写权限。用户也可以设置其他的访问策略，比如对一个对象可以设置公共访问策略，允许所有人对其都有读权限。SSE-KMS方式加密的对象即使设置了ACL，跨租户也不生效。

OBS用户在上传对象时可以设置权限控制策略，也可以通过ACL操作API接口对已存在的对象更改或者获取ACL(access control list)。

对象访问权限与桶访问权限类似（参见[桶访问权限](#)），也可支持预定义访问策略或直接设置。

对象访问权限（[ACL](#)）可以通过三种方式设置：

1. 上传对象时指定预定义访问策略，参见[代码示例一：上传对象时指定预定义访问策略](#)。
2. 调用ObsClient.setObjectAcl指定预定义访问策略，参见[代码示例二：为对象设置预定义访问策略](#)。
3. 调用ObsClient.setObjectAcl直接设置，参见[代码示例三：直接设置对象访问权限](#)。

接口约束

- 您必须是桶拥有者或拥有设置对象ACL的权限，才能设置对象ACL。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObjectAcl权限，如果使用桶策略则需授予PutObjectAcl权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

- 一个对象的ACL最多支持配置100条授权策略。

方法定义

ObsClient.setObjectAcl(params)

请求参数

表 8-15 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。• 桶命名规则如下：<ul style="list-style-type: none">- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。- 禁止使用IP地址。- 禁止以“-”或“.”开头及结尾。- 禁止两个“.”相邻（如：“my..bucket”）。- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。• 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如： G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
ACL	AclType	可选	<p>参数解释: 预定义访问策略。</p> <p>约束限制: Owner、Grants与ACL不能全为空。</p> <p>取值范围: AclType取值详见AclType。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Owner	Owner	可选	<p>参数解释: 桶的所有者。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • Owner和Grants必须配套使用，且与ACL互斥。 • Owner、Grants与ACL不能全为空。 <p>取值范围: 详见Owner。</p> <p>默认取值: 无</p>
Delivered	boolean	可选	<p>参数解释: 对象ACL是否继承桶的ACL。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • true: 对象ACL继承桶的ACL。 • false: 对象ACL不继承桶的ACL。 <p>默认取值: 无</p>
Grants	Grant[]	可选	<p>参数解释: 被授权用户权限信息。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • Owner和Grants必须配套使用，且与ACL互斥。 • Owner、Grants与ACL不能全为空。 <p>取值范围: 详见Grant。</p> <p>默认取值: 无</p>

表 8-16 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限，其他任何人都没有访问权限。

常量名	原始值	说明
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	桶公共读，桶内对象公共读。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadDelivered不能应用于对象。
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	桶公共读写，桶内对象公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadWriteDelivered不能应用于对象。

常量名	原始值	说明
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	<p>设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。</p> <p>默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。</p>

表 8-17 Owner

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时必选	<p>参数解释： 所有者的账号ID，即domain_id。</p> <p>取值范围： 如何获取账号ID请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值： 无</p>
Display Name	string	可选	<p>参数解释： 所有者的账号用户名。</p> <p>默认取值： 无</p>

表 8-18 Grant

参数名称	参数类型	是否必选	描述
Grantee	Grantee	作为请求参数时必选	<p>参数解释： 被授权用户相关信息，详见Grantee。</p>

参数名称	参数类型	是否必选	描述
Permission	PermissionType	作为请求参数时必选	<p>参数解释: 被授予的权限。</p> <p>取值范围: 权限取值范围详见PermissionType。</p> <p>默认取值: 无</p>

表 8-19 Grantee

参数名称	参数类型	是否必选	描述
Type	string	作为请求参数时必选	<p>参数解释: 被授权用户的类型。</p> <p>取值范围: 被授权用户类型的取值范围详见GranteeType。</p> <p>默认取值: 无</p>
ID	string	作为请求参数时，如果Type为用户类型则必选。	<p>参数解释: 被授权用户的账号ID，即domain_id。</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
Name	string	作为请求参数时可选	<p>参数解释: 被授权用户的账号名。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 不支持中文。 只能以字母开头。 长度为6-32个字符。 只能包含英文字母、数字或特殊字符(-_)。 <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
URI	Group UriType	作为请求参数时，如果Type为用户组类型则必选。	参数解释： 被授权的用户组。 取值范围： 授权用户组取值范围详见 GroupUriType 。 默认取值： 无

表 8-20 GranteeType

常量值	说明
Group	授权给用户组。
CanonicalUser	授权给单个用户。

表 8-21 GroupUriType

常量名	原始值	说明
ObsClient.enums.GroupAllUsers	AllUsers	所有用户。
ObsClient.enums.GroupAuthenticatedUsers	AuthenticatedUsers	授权用户，已废弃。
ObsClient.enums.GroupLogDelivery	LogDelivery	日志投递组，已废弃。

表 8-22 PermissionType

常量名	原始值	说明
ObsClient.enums.PermissionRead	READ	如果有桶的读权限，则可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 如果有对象的读权限，则可以获取该对象内容和元数据。
ObsClient.enums.PermissionWrite	WRITE	如果有桶的写权限，则可以上传、覆盖和删除该桶内任何对象和段。 此权限在对象上不适用。

常量名	原始值	说明
ObsClient.enums.PermissionReadAcp	READ_ACP	如果有读ACP的权限，则可以获取对应的桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有读对应桶或对象ACP的权限。
ObsClient.enums.PermissionWriteAcp	WRITE_ACP	如果有写ACP的权限，则可以更新对应桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。 拥有了写ACP的权限，由于可以更改权限控制策略，实际上意味着拥有了完全访问的权限。
ObsClient.enums.PermissionFullControl	FULL_CONTROL	如果有桶的完全控制权限意味着拥有PermissionRead、PermissionWrite、PermissionReadAcp和PermissionWriteAcp的权限。 如果有对象的完全控制权限意味着拥有PermissionRead、PermissionReadAcp和PermissionWriteAcp的权限。

返回结果

表 8-23 返回结果

参数类型	描述
<p>表8-24</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表8-24。</p>

表 8-24 Response

参数名称	参数类型	描述
CommonMsg	ICCommonMsg	<p>参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICCommonMsg。</p>
InterfaceResult	表8-26	<p>参数解释： 操作成功后的结果数据，详见表8-26。</p> <p>约束限制： 当Status大于300时为空。</p>

表 8-25 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 8-26 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例一：上传对象时指定预定义访问策略

以下代码展示如何在上传对象时指定预定义访问策略：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
```

```
secret_access_key: process.env.SECRET_ACCESS_KEY,  
// 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露  
// 风险。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  
// security_token: process.env.SECURITY_TOKEN,  
// endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写  
server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
async function putObject() {  
  try {  
    const params = {  
      // 指定存储桶名称  
      Bucket: "examplebucket",  
      // 指定对象，此处以 example/objectname 为例  
      Key: "example/objectname",  
      // 指定文本对象  
      Body: 'Hello OBS',  
      // 设置对象访问权限为公共读  
      ACL: obsClient.enums.AclPublicRead  
    };  
    // 上传对象  
    const result = await obsClient.putObject(params);  
    if (result.CommonMsg.Status <= 300) {  
      console.log("Put object(%s) under the bucket(%s) successful!!!", params.Key, params.Bucket);  
      console.log("RequestId: %s", result.CommonMsg.RequestId);  
      console.log("StorageClass:%s, ETag:%s", result.InterfaceResult.StorageClass, result.InterfaceResult.ETag);  
      return;  
    }  
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
    console.log("Status: %d", result.CommonMsg.Status);  
    console.log("Code: %s", result.CommonMsg.Code);  
    console.log("Message: %s", result.CommonMsg.Message);  
    console.log("RequestId: %s", result.CommonMsg.RequestId);  
  } catch (error) {  
    console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
    console.log(error);  
  }  
};  
  
putObject();
```

代码示例二：为对象设置预定义访问策略

以下代码展示如何为对象设置预定义访问策略：

```
// 引入obs库  
// 使用npm安装  
const ObsClient = require("esdk-obs-nodejs");  
// 使用源码安装  
// var ObsClient = require('./lib/obs');  
  
// 创建ObsClient实例  
const obsClient = new ObsClient({  
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html  
  access_key_id: process.env.ACCESS_KEY_ID,  
  secret_access_key: process.env.SECRET_ACCESS_KEY,  
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露  
  // 风险。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  
  // security_token: process.env.SECURITY_TOKEN,  
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写  
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
async function setObjectAcl() {  
  try {
```

```
const params = {
  // 指定存储桶名称
  Bucket: "examplebucket",
  // 指定对象名, 此处以 example/objectname 为例
  Key: "example/objectname",
  // 设置对象访问权限为私有读写
  ACL: obsClient.enums.AclPrivate
};
// 设置桶ACL
const result = await obsClient.setObjectAcl(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Set Object(%s)'s acl successful with Bucket(%s)!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
setObjectAcl();
```

代码示例三：直接设置对象访问权限

以下代码展示如何直接设置对象访问权限：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setObjectAcl() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名, 此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定版本对象的版本号
      VersionId: 'G001117FCE89978B0000401205D5DC9A',
      // 指定对象的所有者信息
      Owner: { 'ID': 'ownerid' },
      // 指定被授权的用户信息
      Grants: [
        // 为指定用户设置写权限, 此处用户ID为0a03f5833900d3730f13c00f49d5exxx
        { Grantee: { Type: 'CanonicalUser', ID: '0a03f5833900d3730f13c00f49d5exxx' }, Permission:

```

```
obsClient.enums.PermissionWrite },
  // 为所有用户设置读权限
  { Grantee: { Type: 'Group', URI: obsClient.enums.GroupAllUsers }, Permission:
obsClient.enums.PermissionRead },
  ]
};
// 设置版本对象ACL
const result = await obsClient.setObjectAcl(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Set Object(%s)'s acl successful with Bucket(%s)!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
setObjectAcl();
```

8.4 获取对象 ACL(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

用户执行获取对象ACL的操作，返回信息包含指定对象的权限控制列表信息。

接口约束

- 您必须是桶拥有者或拥有获取对象ACL的权限，才能获取对象ACL。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:GetObjectAcl权限，如果使用桶策略则需授予GetObjectAcl权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getObjectAcl(params)
```

请求参数

表 8-27 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如： G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>

返回结果

表 8-28 返回结果

参数类型	描述
<p>表8-29</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表 8-29。</p>

表 8-29 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表8-31	参数解释: 操作成功后的结果数据，详见 表8-31 。 约束限制: 当Status大于300时为空。

表 8-30 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 8-31 GetObjectAclOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
VersionId	string	参数解释: 对象的版本号。例如: G001117FCE89978B0000401205D5DC9A。
Owner	Owner	参数解释: 对象的所有者, 详见 Owner 。
Delivered	string	参数解释: 桶的ACL是否向桶内对象传递。
Grants	Grant[]	参数解释: 被授权用户权限信息, 详见 Grant 。

表 8-32 Owner

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时必选	参数解释: 所有者的账号ID, 即domain_id。 取值范围: 如何获取账号ID请参见 如何获取账号ID和用户ID? (Node.js SDK) 默认取值: 无
Display Name	string	可选	参数解释: 所有者的账号用户名。 默认取值: 无

表 8-33 Grant

参数名称	参数类型	是否必选	描述
Grantee	Grantee	作为请求参数时必选	参数解释: 被授权用户相关信息, 详见 Grantee 。

参数名称	参数类型	是否必选	描述
Permission	PermissionType	作为请求参数时必选	参数解释: 被授予的权限。 取值范围: 权限取值范围详见PermissionType。

表 8-34 Grantee

参数名称	参数类型	是否必选	描述
Type	string	作为请求参数时必选	参数解释: 被授权用户的类型。 取值范围: 被授权用户类型的取值范围详见GranteeType。 默认取值: 无
ID	string	作为请求参数时, 如果Type为用户类型则必选。	参数解释: 被授权用户的账号ID, 即domain_id。 取值范围: 如何获取账号ID请参见 如何获取账号ID和用户ID? (Node.js SDK) 默认取值: 无
Name	string	作为请求参数时可选	参数解释: 被授权用户的账号名。 约束限制: <ul style="list-style-type: none">不支持中文。只能以字母开头。长度为6-32个字符。只能包含英文字母、数字或特殊字符(-_)。 默认取值: 无

参数名称	参数类型	是否必选	描述
URI	Group UriType	作为请求参数时，如果Type为用户组类型则必选。	参数解释： 被授权的用户组。 取值范围： 授权用户组取值范围详见 GroupUriType 。 默认取值： 无

表 8-35 GranteeType

常量值	说明
Group	授权给用户组。
CanonicalUser	授权给单个用户。

表 8-36 GroupUriType

常量名	原始值	说明
ObsClient.enums.GroupAllUsers	AllUsers	所有用户。
ObsClient.enums.GroupAuthenticatedUsers	AuthenticatedUsers	授权用户，已废弃。
ObsClient.enums.GroupLogDelivery	LogDelivery	日志投递组，已废弃。

表 8-37 PermissionType

常量名	原始值	说明
ObsClient.enums.PermissionRead	READ	如果有桶的读权限，则可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 如果有对象的读权限，则可以获取该对象内容和元数据。
ObsClient.enums.PermissionWrite	WRITE	如果有桶的写权限，则可以上传、覆盖和删除该桶内任何对象和段。 此权限在对象上不适用。

常量名	原始值	说明
ObsClient.enums.PermissionReadAcp	READ_ACP	如果有读ACP的权限，则可以获取对应的桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有读对应桶或对象ACP的权限。
ObsClient.enums.PermissionWriteAcp	WRITE_ACP	如果有写ACP的权限，则可以更新对应桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。 拥有了写ACP的权限，由于可以更改权限控制策略，实际上意味着拥有了完全访问的权限。
ObsClient.enums.PermissionFullControl	FULL_CONTROL	如果有桶的完全控制权限意味着拥有PermissionRead、PermissionWrite、PermissionReadAcp和PermissionWriteAcp的权限。 如果有对象的完全控制权限意味着拥有PermissionRead、PermissionReadAcp和PermissionWriteAcp的权限。

代码示例

本示例用于获取名为examplebucket桶下example/objectname对象的ACL权限控制列表信息。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObjectAcl() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: 'example/objectname',
    };
    // 获取版本对象ACL
    const result = await obsClient.getObjectAcl(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s)'s acl successful with bucket(%s)!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
    }
  }
}
```

```
console.log('Owner[ID]: %s', result.InterfaceResult.Owner.ID);
console.log('Owner[Name]: %s', result.InterfaceResult.Owner.Name);
for (let i = 0; i < result.InterfaceResult.Grants.length; i++) {
  const grant = result.InterfaceResult.Grants[i];
  fmt.Printf("Grant[%d]-Type:%s, ID:%s, URI:%s, Permission:%s\n",
    index, grant.Grantee.Type, grant.Grantee.ID, grant.Grantee.URI, grant.Permission)
};
return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};
getObjectAcl();
```

8.5 列举桶内对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

调用列举桶内对象接口，可列举指定桶内的部分或所有对象的描述信息。您还可以通过设置前缀、数量、起始位置等参数，返回符合您筛选条件的对象信息。返回结果以对象名的字典序排序。

接口约束

- 每次接口调用最多返回1000个对象信息。
- 您必须是桶拥有者或拥有列举桶内对象的权限，才能列举桶内对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:ListBucket权限，如果使用桶策略则需授予ListBucket权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.listObjects(params)
```

请求参数

表 8-38 ListObjectsInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my.bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>
Marker	string	可选	<p>参数解释： 列举桶内对象列表时，指定一个标识符，从该标识符以后按对象名的字典顺序返回对象列表。</p> <p>约束限制： 仅用于非多版本列举。</p> <p>取值范围： 长度大于0且不超过1024的字符串。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
Prefix	string	可选	<p>参数解释: 限定返回的对象名必须带有指定前缀。 例如，假设您拥有以下对象：logs/day1、logs/day2、logs/day3和ExampleObject.jpg。如果您将logs/指定为前缀，将返回以字符串“logs/”开头的三个对象。如果您指定空的前缀且请求中没有其他过滤条件，将返回桶中的所有对象。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
MaxKeys	number	可选	<p>参数解释: 列举对象的最大数目，返回的对象列表将是按照字典顺序的最多前MaxKeys个对象。</p> <p>约束限制: 无</p> <p>取值范围: 1~1000，当超出范围时，按照默认的1000进行处理。</p> <p>默认取值: 1000</p>

参数名称	参数类型	是否必选	描述
Delimiter	string	可选	<p>参数解释:</p> <p>对象名按照此标识符进行分组。通常与prefix参数搭配使用, 如果指定了prefix, 从prefix到第一次出现delimiter间具有相同字符串的对象名会被分成一组, 形成一条CommonPrefixes; 如果没有指定prefix, 从对象名的首字符到第一次出现delimiter间具有相同字符串的对象名会被分成一组, 形成一条CommonPrefixes。</p> <p>例如, 桶中有3个对象, 分别为abcd、abcde、bbcde。如果指定delimiter为d, prefix为a, abcd、abcde会被分成一组, 形成一条前缀为abcd的CommonPrefixes; 如果只指定delimiter为d, abcd、abcde会被分成一组, 形成一条前缀为abcd的CommonPrefixes, 而bbcde会被单独分成一组, 形成一条前缀为bbcd的CommonPrefixes。</p> <p>对于并行文件系统, 不携带此参数时默认列举是递归列举此目录下所有内容, 会列举子目录。在大数据场景下(目录层级深、目录下文件多)的列举, 建议设置[delimiter=/], 只列举当前目录下的内容, 不列举子目录, 提高列举效率。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
Encoding Type	string	可选	<p>参数解释:</p> <p>对响应中的部分元素进行指定类型的编码。如果Delimiter、Marker、Prefix、CommonPrefixes、NextMarker和Key包含xml 1.0标准不支持的控制字符(特殊字符), 需设置该参数为url编码。</p> <p>约束限制: 无</p> <p>取值范围: 可选值为url。</p> <p>默认取值: 无, 不设置则不编码。</p>

返回结果

表 8-39 返回结果

参数类型	描述
表8-40 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见 表 8-40 。

表 8-40 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表8-42	参数解释： 操作成功后的结果数据，详见 表8-42 。 约束限制： 当Status大于300时为空。

表 8-41 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释： OBS服务端返回的HTTP状态码。 取值范围： 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 。
Code	string	参数解释： OBS服务端返回的错误码。
Message	string	参数解释： OBS服务端返回的错误描述。
HostId	string	参数解释： OBS服务端返回的请求服务端ID。
RequestId	string	参数解释： OBS服务端返回的请求ID。

Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 8-42 ListObjectsOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
Location	string	参数解释: 桶所在的区域。
Bucket	string	参数解释: 桶名。
Prefix	string	参数解释: 对象名的前缀，与请求中的该参数对应。 例如，假设您拥有以下对象：logs/day1、logs/day2、logs/day3和ExampleObject.jpg。如果您将logs/指定为前缀，将返回以字符串“logs/”开头的三个对象。如果您指定空的前缀且请求中没有其他过滤条件，将返回桶中的所有对象。
Marker	string	参数解释: 列举对象的起始位置，与请求中的该参数对应。列举桶内对象列表时，指定一个标识符，从该标识符以后按对象名的字典顺序返回对象列表。

参数名称	参数类型	描述
Delimiter	string	<p>参数解释:</p> <p>对象名按照此标识符进行分组。通常与 prefix 参数搭配使用, 如果指定了 prefix, 从 prefix 到第一次出现 delimiter 间具有相同字符串的对象名会被分成一组, 形成一条 CommonPrefixes; 如果没有指定 prefix, 从对象名的首字符到第一次出现 delimiter 间具有相同字符串的对象名会被分成一组, 形成一条 CommonPrefixes。</p> <p>例如, 桶中有3个对象, 分别为abcd、abcde、bbcde。如果指定 delimiter 为 d, prefix 为 a, abcd、abcde 会被分成一组, 形成一条前缀为 abcd 的 CommonPrefixes; 如果只指定 delimiter 为 d, abcd、abcde 会被分成一组, 形成一条前缀为 abcd 的 CommonPrefixes, 而 bbcde 会被单独分成一组, 形成一条前缀为 bbcde 的 CommonPrefixes。</p> <p>对于并行文件系统, 不携带此参数时默认列举是递归列举此目录下所有内容, 会列举子目录。在大数据场景下(目录层级深、目录下文件多)的列举, 建议设置 [delimiter=/], 只列举当前目录下的内容, 不列举子目录, 提高列举效率。</p>
MaxKeys	number	<p>参数解释:</p> <p>列举对象的最大数目, 与请求中的该参数对应。返回的对象列表将是按照字典顺序的最多前 MaxKeys 个对象。</p>
IsTruncated	string	<p>参数解释:</p> <p>表明本次请求是否返回了全部结果。因为每次列举返回对象的数量上限是1000个, 如果对象个数大于1000, 则无法通过一次请求返回全部结果。</p> <p>取值范围:</p> <ul style="list-style-type: none">• true: 表示没有返回全部结果。• false: 表示已返回了全部结果。

参数名称	参数类型	描述
NextMarker	string	参数解释: 下次列举对象请求的起始位置。如果本次没有返回全部结果，响应请求中将包含此字段，用于标明本次请求列举到的最后一个对象。后续请求可以指定Marker参数等于该值来列举剩余的对象。
Contents	Content []	参数解释: 桶内对象列表，详见 Content 。
CommonPrefixes	CommonPrefix []	参数解释: 当请求中设置了Delimiter分组字符时，返回按Delimiter分组后的对象名称前缀列表。 取值范围: 详见 CommonPrefix 。
EncodingType	string	参数解释: 对响应中的部分元素进行指定类型的编码。如果Delimiter、Marker、Prefix、CommonPrefixes、NextMarker和Key包含xml 1.0标准不支持的控制字符（特殊字符），需设置该参数为url编码。

表 8-43 Content

参数名称	参数类型	描述
Key	string	参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。
LastModified	string	参数解释: 对象最近一次被修改的时间（UTC时间）。

参数名称	参数类型	描述
ETag	string	参数解释: 对象的base64编码的128位MD5摘要。ETag是对象内容的唯一标识,可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A,下载对象时ETag为B,则说明对象内容发生了变化。ETag只反映变化的内容,而不是其元数据。上传的对象或复制操作创建的对象,都有唯一的ETag。 约束限制: 当对象是服务端加密的对象时,ETag值不是对象的MD5值。
Size	number	参数解释: 对象的字节数。
Owner	Owner	参数解释: 对象的所有者,包含对象拥有者DomainId和对象拥有者名称,详见 Owner 。
StorageClass	string	参数解释: 对象的存储类型。
Type	string	参数解释: 对象的类型。 取值范围: <ul style="list-style-type: none">• NORMAL: 普通对象• APPENDABLE: 追加写对象

表 8-44 CommonPrefix

参数名称	参数类型	描述
Prefix	string	参数解释: 为桶内对象的前缀。 取值范围: 长度大于0且不超过1024的字符串。 默认取值: 无

表 8-45 Owner

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时必选	<p>参数解释: 所有者的账号ID, 即domain_id。</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
Display Name	string	可选	<p>参数解释: 所有者的账号用户名。</p> <p>默认取值: 无</p>

代码示例一：简单列举

以下代码展示如何简单列举对象，最多返回1000个对象：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
    };
    // 列举桶内对象
    const result = await obsClient.listObjects(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        const val = result.InterfaceResult.Contents[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
          j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
      }
    }
    return;
  }
}
```



```
};  
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
console.log("Status: %d", result.CommonMsg.Status);  
console.log("Code: %s", result.CommonMsg.Code);  
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
    console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
    console.log(error);  
};  
};  
};  
listObjects();
```

📖 说明

- 每次至多返回1000个对象，如果指定桶包含的对象数量大于1000，则返回结果中InterfaceResult.IsTruncated为true表明本次没有返回全部对象，并可通过InterfaceResult.NextMarker获取下次列举的起始位置。
- 如果想获取指定桶包含的所有对象，可以采用[代码示例五：分页列举全部对象](#)的方式。

代码示例二：指定数目列举

以下代码展示如何指定数目列举对象：

```
// 引入obs库  
// 使用npm安装  
const ObsClient = require("esdk-obs-nodejs");  
// 使用源码安装  
// var ObsClient = require('./lib/obs');  
  
// 创建ObsClient实例  
const obsClient = new ObsClient({  
    // 推荐通过环境变量获取AKSK，这里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  
    access_key_id: process.env.ACCESS_KEY_ID,  
    // 您可以登录访问管理控制台获取访问密钥AK/SK，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html  
    secret_access_key: process.env.SECRET_ACCESS_KEY,  
    // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露  
    // 风险。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  
    // security_token: process.env.SECURITY_TOKEN,  
    // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写  
    server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
async function listObjects() {  
    try {  
        const params = {  
            // 指定存储桶名称  
            Bucket: "examplebucket",  
            // 指定返回的最大对象数，此处以 100 为例，返回的对象列表将是按照字典顺序的最多前max-keys个对象，  
            // 默认值为1000  
            MaxKeys: 100,  
        };  
        // 列举桶内对象  
        const result = await obsClient.listObjects(params);  
        if (result.CommonMsg.Status <= 300) {  
            console.log("List objects under the bucket(%s) successful!", params.Bucket);  
            console.log("RequestId: %s", result.CommonMsg.RequestId);  
            for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {  
                const val = result.InterfaceResult.Contents[j];  
                console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',  
                    j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);  
            };  
            return;  
        };  
    };  
};
```

```
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

listObjects();
```

代码示例三：指定前缀列举

以下代码展示如何指定前缀列举对象：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');
// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定列举对象前缀，此处以“test/”前缀为例，满足指定前缀的对象会被列举
      Prefix: "test/",
      // 指定返回的最大对象数，此处以 100 为例，返回的对象列表将是按照字典顺序的最多前max-keys个对象，默认值为1000
      MaxKeys: 100,
    };

    // 列举桶内对象
    const result = await obsClient.listObjects(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        const val = result.InterfaceResult.Contents[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
          j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
      };
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
```

```
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

listObjects();
```

代码示例四：指定起始位置列举

以下代码展示如何指定起始位置列举对象：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定列举对象前缀，此处以“test/”前缀为例，满足指定前缀的对象会被列举
      Prefix: "test/",
      // 指定返回的最大对象数，此处以 100 为例，返回的对象列表将是按照字典顺序的最多前max-keys个对象，默认值为1000
      MaxKeys: 100,
      // 指定列举对象的起始位置，此处以“test/test2”为例
      Marker: "test/test2",
    };
    // 列举桶内对象
    const result = await obsClient.listObjects(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        const val = result.InterfaceResult.Contents[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
          j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
      };
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};
```

```
listObjects();
```

代码示例五：分页列举全部对象

以下代码展示分页列举全部对象：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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
  // 您可以通过登录访问管理控制台获取访问密钥AK/SK，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定编码方式，此处以“url”为例，如果列举对象中存在特殊字符，则该参数必传
      EncodingType: "url",
    };
    while (true) {
      // 列举桶内版本对象
      const result = await obsClient.listObjects(params);
      if (result.CommonMsg.Status > 300) {
        console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
        console.log("Status: %d", result.CommonMsg.Status);
        console.log("Code: %s", result.CommonMsg.Code);
        console.log("Message: %s", result.CommonMsg.Message);
        console.log("RequestId: %s", result.CommonMsg.RequestId);
        return;
      };
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        const val = result.InterfaceResult.Contents[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
          j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
      };
      if (result.InterfaceResult.IsTruncated === "true") {
        params.KeyMarker = result.InterfaceResult.NextKeyMarker;
      } else {
        break;
      };
    };
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

listObjects();
```

代码示例六：列举文件夹中的所有对象

OBS本身是没有文件夹的概念的，桶中存储的元素只有对象。文件夹对象实际上是一个大小为0且对象名以“/”结尾的对象，将这个文件夹对象名作为前缀，即可模拟列举文件夹中对象的功能。以下代码展示如何列举文件夹中的对象：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定列举对象前缀，此处以“test/”前缀为例，满足指定前缀的对象会被列举
      Prefix: "test/",
      // 指定编码方式，此处以“url”为例，如果列举对象中存在特殊字符，则该参数必传
      EncodingType: "url",
    };
    let count = 1
    while (true) {
      // 列举桶内版本对象
      const result = await obsClient.listObjects(params);
      if (result.CommonMsg.Status > 300) {
        console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
        console.log("Status: %d", result.CommonMsg.Status);
        console.log("Code: %s", result.CommonMsg.Code);
        console.log("Message: %s", result.CommonMsg.Message);
        console.log("RequestId: %s", result.CommonMsg.RequestId);
        return;
      }
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        const val = result.InterfaceResult.Contents[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
          count++, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
      }
      if (result.InterfaceResult.IsTruncated === "true") {
        params.Marker = result.InterfaceResult.NextMarker;
      } else {
        break;
      }
    }
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};
```

```
listObjects();
```

8.6 删除对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

说明

请您谨慎使用删除操作，如果对象所在的桶未开启多版本控制功能，该对象一旦删除将无法恢复。

功能说明

为节省空间和成本，您可以根据需要删除指定桶中的单个对象。

接口约束

- 您必须是桶拥有者或拥有删除对象的权限，才能删除对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:DeleteObject权限，如果使用桶策略则需授予DeleteObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 桶没有开启多版本控制功能时，已删除的对象不可恢复，请谨慎操作。

方法定义

```
ObsClient.deleteObject(params)
```

请求参数

表 8-46 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如： G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>

返回结果

表 8-47 返回结果

参数类型	描述
<p>表8-48</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表 8-48。</p>

表 8-48 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	表8-50	参数解释: 操作成功后的结果数据, 详见 表8-50 。 约束限制: 当Status大于300时为空。

表 8-49 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 8-50 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
DeleteMarker	boolean	参数解释: 标识删除的对象是否是删除标记。 取值范围: <ul style="list-style-type: none"> • true: 是删除标记。 • false: 不是删除标记。
VersionId	string	参数解释: 对象的版本号, 用于删除指定版本号的对象。例如: G001117FCE89978B0000401205D5DC9。

代码示例

本示例用于删除examplebucket桶中的example/objectname对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定删除对象, 此处以 example/objectname 为例
      Key: 'example/objectname',
    };
    // 删除对象
    const result = await obsClient.deleteObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
  }
}
```

```
console.log("Code: %s", result.CommonMsg.Code);  
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
  console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
  console.log(error);  
};  
};  
};  
deleteObject();
```

8.7 批量删除对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

说明

请您谨慎使用批量删除操作，如果对象所在的桶未开启多版本控制功能，该对象一旦删除将无法恢复。

功能说明

为节省空间和成本，您可以根据需要删除指定桶中的多个对象。

批量删除对象特性用于将一个桶内的部分对象一次性删除，删除后不可恢复。批量删除对象要求返回结果里包含每个对象的删除结果。OBS的批量删除对象使用同步删除对象的方式，每个对象的删除结果返回给请求用户。

接口约束

- 您必须是桶拥有者或拥有批量删除对象的权限，才能批量删除对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:DeleteObject权限，如果使用桶策略则需授予DeleteObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 桶没有开启多版本控制功能时，已删除的对象不可恢复，请谨慎操作。
- 批量删除对象一次能接收最大对象数目为1000个，如果超出限制，服务端会返回请求不合法。
- 并发任务分配后，在循环删除多个对象过程中，如果发生内部错误，有可能出现数据不一致的情况（某个对象索引数据删除但还有元数据）。

方法定义

```
ObsClient.deleteObjects(params)
```

请求参数

表 8-51 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>
Quiet	boolean	可选	<p>参数解释: 批量删除对象的响应方式。</p> <p>约束限制: 无</p> <p>取值范围:</p> <ul style="list-style-type: none">false: 表示详细模式，返回删除成功和删除失败的所有结果；true: 表示简单模式，只返回删除过程中出错的结果。 <p>默认取值: false</p>

参数名称	参数类型	是否必选	描述
Objects	ObjectToDelete []	必选	参数解释: 待删除的对象列表, 详情参考 ObjectToDelete 。 约束限制: 每次最多删除1000个对象。 取值范围: 无 默认取值: 无
EncodingType	string	可选	参数解释: 用于指定对响应中的Key进行指定类型的编码。如果Key包含xml 1.0标准不支持的控制字符, 可通过设置该参数对响应中的Key进行编码。 约束限制: 无 取值范围: 可选值为url。 默认取值: 无, 不设置则不编码。

表 8-52 ObjectToDelete

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号，用于删除指定版本号的对象。例如： G001117FCE89978B0000401205D5DC9。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无，如果不设置则默认删除最新版本的对象。</p>

返回结果

表 8-53 返回结果

参数类型	描述
<p>表8-54</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表 8-54。</p>

表 8-54 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表8-56	参数解释: 操作成功后的结果数据，详见 表8-56 。 约束限制: 当Status大于300时为空。

表 8-55 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 8-56 DeleteObjectsOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
DeletedIds	Deleted []	参数解释: 删除成功的对象列表, 详见 Deleted 。
Errors	Error []	参数解释: 删除失败的对象列表, 详见 Error 。
EncodingType	string	参数解释: 用于指定对响应中的Key进行指定类型的编码。如果Key包含xml 1.0标准不支持的控制字符, 可通过设置该参数对响应中的Key进行编码。

表 8-57 Deleted

参数名称	参数类型	描述
Key	string	参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径, 路径中不包含桶名。 例如, 您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中, 对象名为folder/test.txt。
VersionId	string	参数解释: 对象的版本号, 用于删除指定版本号的对象。 例如: G001117FCE89978B0000401205D5DC9。
DeleteMarker	bool	参数解释: 标识删除的对象是否是删除标记。 取值范围: <ul style="list-style-type: none">• true: 是删除标记。• false: 不是删除标记。

参数名称	参数类型	描述
DeleteMarkerVersionId	string	<p>参数解释: 请求创建或删除的删除标记版本号。</p> <p>当批量删除请求访问的桶是多版本桶时，如果创建或删除一个删除标记，响应消息会返回该元素。该元素在以下两种情况中会出现：</p> <ul style="list-style-type: none"> • 用户发送不带版本删除请求，即请求只有对象名，无版本号。这种情况下，系统会创建一个删除标记，并在响应中返回该删除标记的版本号。 • 用户发送带版本删除请求，即请求同时包含对象名以及版本号，但是该版本号标识一个删除标记。这种情况下，系统会删除此删除标记，并在响应中返回该删除标记的版本号。

表 8-58 Error

参数名称	参数类型	描述
Key	string	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p>
VersionId	string	<p>参数解释: 对象的版本号，用于删除指定版本号的对象。 例如： G001117FCE89978B0000401205D5DC9。</p>
Code	string	<p>参数解释: 删除失败的错误码。可参考OBS错误码。</p>
Message	string	<p>参数解释: 删除失败的错误原因。可参考OBS错误码。</p>

代码示例

以下代码展示了如何进行批量删除对象：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
```

```
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteObjects() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定删除列表
      Objects: [
        { Key: 'objectname1', VersionId: "version1" },
        { Key: 'objectname2', VersionId: "version2" },
        { Key: 'objectname3', VersionId: "version3" }
      ]
    };
    // 批量删除对象
    const result = await obsClient.deleteObjects(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      // 获取删除成功的对象
      console.log("Deleted:");
      for (let i = 0; i < result.InterfaceResult.Deleted.length; i++) {
        const deleted = result.InterfaceResult.Deleted[i];
        console.log("Deleted[%d]-Key:%s, VersionId:%s", i, deleted.Key, deleted.VersionId);
      };
      // 获取删除失败的对象
      console.log("Errors:");
      for (let i = 0; i < result.InterfaceResult.Errors.length; i++) {
        const err = result.InterfaceResult.Errors[i];
        console.log("Errors[%d]-Key:%s, Code:%s", i, err.Key, err.Code);
      };
      console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
      console.log("Status: %d", result.CommonMsg.Status);
      console.log("Code: %s", result.CommonMsg.Code);
      console.log("Message: %s", result.CommonMsg.Message);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
    } catch (error) {
      console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
      console.log(error);
    };
  };
};

deleteObjects();
```

8.8 复制对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

用户可以根据需要将存储在OBS上的对象复制到指定路径下。复制对象操作将创建需要复制的对象的副本。在单次操作中，您可以创建最大5GB的对象副本。

接口约束

- 您必须是桶所有者或拥有复制对象的权限，才能复制对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObject权限，如果使用桶策略则需授予PutObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 目标对象大小范围是[0, 5GB]，如果源对象大小超过5GB，只能使用[复制段\(Node.js SDK\)](#)功能复制部分对象。

方法定义

```
ObsClient.copyObject(params)
```

请求参数

表 8-59 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p>
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
CopySource	string	必选	<p>参数解释: 用来指定复制对象操作的源桶名以及源对象名。当源对象存在多个版本时，通过 versionId 参数指定版本源对象。</p> <p>示例: source_bucket/sourceObject?versionId=G001117FCE89978B0000401205D5DC9A</p> <p>约束限制: 无</p> <p>约束限制: 中文字符与%，需要进行URL编码。</p> <p>默认取值: 无</p>
ACL	AclType	可选	<p>参数解释: 复制对象时可指定的预定义访问策略。有关访问控制列表（Access Control List, ACL）功能的详细信息可参见ACL功能介绍。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的访问策略选项参见AclType。</p> <p>默认取值: 无</p>
MetadataDirective	MetadataDirectiveType	可选	<p>参数解释: 复制源对象属性的策略。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的对象属性选项参见MetadataDirectiveType。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
CopySourceIfMatch	string	可选	参数解释: 如果源对象的ETag值与该参数值相同, 则进行复制, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无
CopySourceIfNoneMatch	string	可选	参数解释: 如果源对象的ETag值与该参数值不相同, 则进行复制, 否则返回错误。 约束限制: 无 取值范围: 长度为32的字符串。 默认取值: 无
CopySourceIfUnmodifiedSince	string	可选	参数解释: 如果源对象在指定的时间后没有修改, 则进行复制, 否则抛出异常。 约束限制: 该参数值必须符合 http://www.ietf.org/rfc/rfc2616.txt 规定的HTTP时间格式。 取值范围: 无 默认取值: 无

参数名称	参数类型	是否必选	描述
CopySourceIfModifiedSince	string	可选	<p>参数解释: 如果源对象在指定的时间后有修改, 则进行复制, 否则抛出异常。</p> <p>约束限制: 该参数值必须符合http://www.ietf.org/rfc/rfc2616.txt规定的HTTP时间格式。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
WebsiteRedirectLocation	string	可选	<p>参数解释: 当桶设置了Website配置, 可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。</p> <p>例如, 重定向请求到桶内另一对象: WebsiteRedirectLocation:/anotherPage.html</p> <p>或重定向请求到一个外部URL: WebsiteRedirectLocation:http://www.example.com/</p> <p>OBS将这个值从头域中取出, 保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向, 不支持为桶中文件夹下的对象设置重定向。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
CopySourceSseC	string	当使用SSE-C方式时，必选	参数解释： 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。 约束限制： 仅支持AES256。 取值范围： 可选值：AES256，即选择SSE-C方式加密对象。 默认取值： 无
CopySourceSseKey	string	当使用SSE-C方式时，必选	参数解释： SSE-C方式下加密使用的密钥，该密钥用于加密对象。 约束限制： 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVaobncLht/rCB2o/9Cw= 取值范围： 无 默认取值： 无
CacheControl	string	可选	参数解释： 复制时重写响应中的Cache-Control头。 约束限制： 无 取值范围： 参见HTTP标准头域Cache-Control的取值。 默认取值： 无

参数名称	参数类型	是否必选	描述
ContentDisposition	string	可选	参数解释: 复制时重写响应中的Content-Disposition头。 约束限制: 无 取值范围: 参见HTTP标准头域Content-Disposition的取值。 默认取值: 无
ContentEncoding	string	可选	参数解释: 复制时重写响应中的Content-Encoding头。 约束限制: 无 取值范围: 参见HTTP标准头域Content-Encoding的取值。 默认取值: 无
ContentLanguage	string	可选	参数解释: 复制时重写响应中的Content-Language头。 约束限制: 无 取值范围: 参见HTTP标准头域Content-Language的取值。 默认取值: 无

参数名称	参数类型	是否必选	描述
ContentType	string	可选	参数解释: 复制时重写响应中的Content-Type头。 约束限制: 无 取值范围: 常见的Content-Type (MIME) 列表参见 如何理解Content-Type (MIME) (Node.js SDK) 默认取值: 无
Expires	string	可选	参数解释: 复制时重写响应中的Expires头。 约束限制: 无 取值范围: 参见HTTP标准头域Expires的取值。 默认取值: 无
StorageClass	StorageClassType	可选	参数解释: 复制时设置对象的存储类别。 约束限制: 无 取值范围: 可选择的存储类型参见 StorageClassType 。 默认取值: 无, 如果未设置此头域, 则对象的存储类型默认为桶的存储类型。

参数名称	参数类型	是否必选	描述
Metadata	object	可选	<p>参数解释: 目标对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据，以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时，加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为：每个键和值的UTF-8 编码中的字节总数。 • 自定义元数据的key值不区分大小写，OBS统一转为小写进行存储。value值区分大小写。 • 自定义元数据key-value对都必须符合US-ASCII。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。 <p>取值范围: 无</p> <p>默认取值: 无</p>
GrantRead	string	可选	<p>参数解释: 被授予READ权限的账号的账号ID，即domain_id。账号被授予READ权限后，针对当前正在复制的对象，允许获取对象的内容、获取对象的元数据。</p> <p>约束限制: 如果授权给多个租户，需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantReadAcp	string	可选	<p>参数解释: 被授予READ_ACP权限的账号的账号ID, 即domain_id。账号被授予READ_ACP权限后, 针对当前正在复制的对象, 允许读对象的ACL信息。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantWriteAcp	string	可选	<p>参数解释: 被授予WRITE_ACP权限的账号的账号ID, 即domain_id。账号被授予WRITE_ACP权限后, 针对当前正在复制的对象, 允许修改对象的ACL信息。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantFullControl	string	可选	<p>参数解释: 被授予FULL_CONTROL权限的账号的账号ID，即domain_id。账号被授予FULL_CONTROL权限后，针对当前正在复制的对象，允许进行任何操作，读对象、获取对象元数据、获取对象ACL、写对象ACL。</p> <p>约束限制: 如果授权给多个租户，需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
SuccessActionRedirect	string	可选	<p>参数解释: 此参数的值是一个URL，用于指定当此次请求操作成功响应后的重定向的地址。如果此参数值有效且操作成功，响应码为303，Location头域由此参数以及桶名、对象名、对象的ETag组成。如果此参数值无效，则OBS忽略此参数的作用，Location头域为对象地址，响应码根据操作成功或失败正常返回。</p> <p>约束限制: 无</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

表 8-60 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限，其他任何人都没有访问权限。

常量名	原始值	说明
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	桶公共读，桶内对象公共读。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadDelivered不能应用于对象。
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	桶公共读写，桶内对象公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadWriteDelivered不能应用于对象。

常量名	原始值	说明
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	<p>设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。</p> <p>默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。</p>

表 8-61 MetadataDirectiveType

常量名	原始值	说明
ObsClient.enums.CopyMetadata	COPY	<p>复制对象时，复制源对象属性。</p> <p>须知 该值仅在复制对象(Node.js SDK)接口中使用。</p>
ObsClient.enums.ReplaceMetadata	REPLACE	<p>使用当前请求中携带的头域完整替换，未指定的元数据会被删除。</p>
ObsClient.enums.ReplaceNewMetadata	REPLACE_NEW	<p>对于已经存在值的元数据进行替换，不存在值的元数据进行赋值，未指定的元数据保持不变。</p> <p>须知 该值仅在设置对象元数据(Node.js SDK)接口中使用。</p>

表 8-62 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。</p>

常量名	原始值	说明
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

📖 说明

- 如果包含CopySourceIfUnmodifiedSince并且不符合，或者包含CopySourceIfMatch并且不符合，或者包含CopySourceIfModifiedSince并且不符合，或者包含CopySourceIfNoneMatch并且不符合，抛出异常中HTTP状态码为：412 precondition failed。
- CopySourceIfModifiedSince和CopySourceIfNoneMatch可以一起使用；CopySourceIfUnmodifiedSince和CopySourceIfMatch可以一起使用。

返回结果

表 8-63 返回结果

参数类型	描述
表8-64 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见 表8-64 。

表 8-64 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表8-66	参数解释： 操作成功后的结果数据，详见 表8-66 。 约束限制： 当Status大于300时为空。

表 8-65 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 8-66 CopyObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
LastModified	string	参数解释: 目标对象的最近一次修改时间（UTC时间）。

参数名称	参数类型	描述
Etag	string	<p>参数解释: 目标对象的Etag值。对象的base64编码的128位MD5摘要。Etag是对象内容的唯一标识,可以通过该值识别对象内容是否有变化。比如上传对象时Etag为A,下载对象时Etag为B,则说明对象内容发生了变化。Etag只反映变化的内容,而不是其元数据。上传的对象或复制操作创建的对象,都有唯一的Etag。</p> <p>约束限制: 当对象是服务端加密的对象时,Etag值不是对象的MD5值。</p>
CopySourceVersionId	string	<p>参数解释: 源对象的版本号。</p>
VersionId	string	<p>参数解释: 目标对象的版本号。</p>

代码示例一：简单复制

以下代码展示了如何进行简单复制：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK,这里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,获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS,同样建议您尽量避免使用硬编码,以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint,这里以中国-香港为例,其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyObject() {
  try {
    const params = {
      // 指定目标存储桶名称
      Bucket: "examplebucket",
      // 指定目标对象名,此处以 example/objectname 为例
      Key: 'example/objectname',
      // 指定复制源桶名和源对象名,此处复制源桶名以sourcebucketname为例,复制源对象名以sourceobjectkey为例
      CopySource: 'sourcebucketname/sourceobjectname'
    };
    // 复制对象
    const result = await obsClient.copyObject(params);
    if (result.CommonMsg.Status <= 300) {
```

```
console.log("Copy Object(bucket:%s, object: %s) successful from bucket/object: %s!", params.Bucket,
params.Key, params.CopySource);
console.log("RequestId: %s", result.CommonMsg.RequestId);
console.log("ETag: %s, LastModified:%s", result.InterfaceResult.ETag,
result.InterfaceResult.LastModified);
return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
console.log(error);
};
};
};
copyObject();
```

代码示例二：重写对象属性

📖 说明

使用**Metadata**参数指定待重写的自定义对象属性；使用**MetadataDirective**参数指定重写选项，支持ObsClient.enums.CopyMetadata（从源对象复制）和ObsClient.enums.ReplaceMetadata（重写）两个值。

以下代码展示了如何在复制对象时重写对象属性：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
// 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
// security_token: process.env.SECURITY_TOKEN,
// endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyObject() {
try {
const params = {
// 指定目标存储桶名称
Bucket: "examplebucket",
// 指定目标对象名，此处以 example/objectname 为例
Key: 'example/objectname',
// 指定复制源桶名和源对象名，此处复制源桶名以sourcebucketname为例，复制源对象名以
sourceobjectkey为例
CopySource: 'sourcebucketname/sourceobjectname'
// 指定自定义元数据
Metadata: {'property':'property-value'},
// 指定复制源对象属性的策略，此处以 ObsClient.enums.ReplaceMetadata 为例，含义是重写源对象属性
MetadataDirective: ObsClient.enums.ReplaceMetadata
};
// 复制对象
```

```
const result = await obsClient.copyObject(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Copy Object(bucket:%s, object: %s) successful from bucket/object: %s!", params.Bucket,
params.Key, params.CopySource);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  console.log("ETag: %s, LastModified:%s", result.InterfaceResult.ETag,
result.InterfaceResult.LastModified);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
copyObject();
```

代码示例三：限定条件复制

复制对象时，可以指定一个或多个限定条件，满足限定条件时则进行复制，否则返回异常码，复制对象失败。

📖 说明

- 源对象的ETag值是指源对象数据的MD5校验值。
- 如果包含CopySourceIfUnmodifiedSince并且不符合，或者包含CopySourceIfMatch并且不符合，或者包含CopySourceIfModifiedSince并且不符合，或者包含CopySourceIfNoneMatch并且不符合，则复制失败，返回异常码：412 precondition failed。
- CopySourceIfModifiedSince和CopySourceIfNoneMatch可以一起使用；CopySourceIfUnmodifiedSince和CopySourceIfMatch可以一起使用。

以下代码展示了如何进行限定条件复制：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定复制对象，此处以 example/objectname 为例
    };
  }
}
```

```
Key: 'example/objectname',
// 指定复制源桶名和对象名, 此处复制源桶名以sourcebucketname为例, 复制源对象名以sourceobjectkey
为例
CopySource: 'sourcebucketname/sourceobjectname'
// 指定限定条件, 如果源对象在指定的时间后有修改, 则进行复制
CopySourceIfModifiedSince: 'Thu, 31 Dec 2015 16:00:00 GMT',
// 指定限定条件, 如果源对象的ETag值与该参数值不相同, 则进行复制
CopySourceIfNoneMatch: 'none-match-etag'
};
// 复制对象
const result = await obsClient.copyObject(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Copy Object(bucket:%s, object: %s) successful from bucket/object: %s!", params.Bucket,
params.Key, params.CopySource);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  console.log("ETag: %s, LastModified:%s", result.InterfaceResult.ETag,
result.InterfaceResult.LastModified);
  return;
}
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
copyObject();
```

代码示例四：重写对象访问权限

以下代码展示了如何在复制对象时重写对象访问权限：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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
  // 您可以通过登录访问管理控制台获取访问密钥AK/SK, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露
  // 风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定复制对象, 此处以 example/objectname 为例
      Key: 'example/objectname',
      // 指定复制源桶名和对象名, 此处复制源桶名以sourcebucketname为例, 复制源对象名以sourceobjectkey
      // 为例
      CopySource: 'sourcebucketname/sourceobjectname'
      // 复制时重写对象访问权限为公共读
    };
  }
}
```

```
    ACL : obsClient.enums.AclPublicRead
  };
  // 复制对象
  const result = await obsClient.copyObject(params);
  if (result.CommonMsg.Status <= 300) {
    console.log("Copy Object(bucket:%s, object: %s) successful from bucket/object: %s!", params.Bucket,
params.Key, params.CopySource);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    console.log("ETag: %s, LastModified:%s", result.InterfaceResult.ETag,
result.InterfaceResult.LastModified);
    return;
  };
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
  console.log("Status: %d", result.CommonMsg.Status);
  console.log("Code: %s", result.CommonMsg.Code);
  console.log("Message: %s", result.CommonMsg.Message);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
copyObject();
```

9 多段相关接口(Node.js SDK)

9.1 多段相关接口说明(Node.js SDK)

对于较大文件上传，可以切分成段上传。用户可以在如下的应用场景内（但不仅限于此），使用分段上传的模式：

- 上传超过100MB大小的文件。
- 网络条件较差，和OBS服务端之间的链接经常断开。
- 上传前无法确定将要上传文件的大小。

分段上传分为如下3个步骤：

1. **初始化分段上传任务。**
2. **逐个或并行上传段。**
3. **合并段或取消分段上传任务。**

分段上传的主要目的是解决大文件上传或网络条件较差的情况。下面的代码示例展示了如何使用分段上传并发上传大文件：

```
// 引入obs库
// 使用npm安装
const ObsClient = require('esdk-obs-nodejs');
// 使用源码安装
// var ObsClient = require('./lib/obs');
const fs = require('fs');

// 创建ObsClient实例
var obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// 指定存储桶名称
const Bucket = 'bucketname'
```

```
// 指定对象名, 此处以 example/objectname 为例
const Key = 'objectname'

// 指定分段大小
const DEFAULT_PART_SIZE = 9 * 1024 * 1024;
// 指定并发数为20
const CONCURRENCY = 20

// 准备分段上传的参数
const preparePartParams = (Bucket, Key, UploadId) => (sampleFile, partSize = DEFAULT_PART_SIZE) => {
  try {
    const fileSize = fs.lstatSync(sampleFile).size;
    const partCount = fileSize % partSize === 0 ? Math.floor(fileSize / partSize) : Math.floor(fileSize /
partSize) + 1;

    const uploadPartParams = [];
    // 指定并发上传
    for (let i = 0; i < partCount; i++) {
      // 分段在文件中的起始位置
      let Offset = i * partSize;
      // 分段大小
      let currPartSize = (i + 1 === partCount) ? fileSize - Offset : partSize;
      // 分段号
      let PartNumber = i + 1;
      uploadPartParams.push({
        Bucket,
        Key,
        PartNumber,
        UploadId,
        Offset,
        SourceFile: sampleFile,
        PartSize: currPartSize,
      });
    };

    return ({ uploadPartParams, fileSize });
  } catch (error) {
    console.log(error)
  };
};

/**
 * uploadSuccessSize: 已经上传成功的大小
 * uploadSuccessCount: 已经上传成功的段数量
 * concurrency: 当前并发数
 */
let uploadSuccessSize = 0;
let uploadSuccessCount = 0;
let concurrency = 0

const parts = [];

// 上传段
const uploadPart = (uploadPartParam, otherUploadPartInfo) => {
  const partCount = otherUploadPartInfo.partCount;
  const fileSize = otherUploadPartInfo.fileSize;
  concurrency++;
  return obsClient
    .uploadPart(uploadPartParam)
    .then(result => {
      const { PartNumber, PartSize } = uploadPartParam;
      if (result.CommonMsg.Status < 300) {
        uploadSuccessCount++;
        uploadSuccessSize += PartSize;
        // 打印上传进度
        console.log(`the current concurrent count is ${concurrency} | uploaded segment: $
{uploadSuccessCount}/${partCount}. the progress is ${{(uploadSuccessSize / fileSize) * 100}.toFixed(2)}% |
the partNumber ${PartNumber} upload succeeded.`);
      }
    });
};
```



```
        parts.push({ PartNumber, ETag: result.InterfaceResult.ETag });
    } else {
        console.log(result.CommonMsg.Code, parts);
    };
    concurrency--;
  }).catch(function (err) {
    console.log(err);
    throw err;
  })
};

// 分段上传
const uploadFile = (sourceFile) => {
  // 初始化分段任务
  obsClient.initiateMultipartUpload({
    Bucket,
    Key
  }).then(res => {
    const Status = res.CommonMsg.Status;
    const UploadId = res.InterfaceResult.UploadId;

    if (typeof Status === 'number' && Status > 300) {
      console.log(`initiateMultipartUpload failed! Status:${Status}`);
      return;
    };

    const partParams = preparePartParams(Bucket, Key, UploadId)(sourceFile);
    const uploadPartParams = partParams.uploadPartParams;
    const fileSize = partParams.fileSize;
    const partCount = uploadPartParams.length;
    const otherUploadPartInfo = { fileSize, partCount };

    // 调用并行上传函数
    parallelFunc(uploadPartParams, (param) => uploadPart(param, otherUploadPartInfo),
CONCURRENCY)
      .then(() => {
        // 合并段
        obsClient.completeMultipartUpload({
          Bucket,
          Key,
          UploadId,
          Parts: parts.sort((a, b) => a.PartNumber - b.PartNumber)
        }, (err, result) => {
          if (err) {
            console.log('Error-->' + err);
          } else {
            console.log('Status-->' + result.CommonMsg.Status);
          };
        });
      });
  }).catch(function (err) {
    console.log(err)
  });
};

/**
 * 实现函数并行执行
 * @param {Array} params 回调函数的参数数组
 * @param {Promise} promiseFn 回调函数
 * @param {number} limit 并行数
 */
const parallelFunc = (params, promiseFn, limit) => {
  return new Promise((resolve) => {
    let concurrency = 0;
    let finished = 0;
    const count = params.length;
    const run = (param) => {
```

```
concurrency++;
promiseFn(param)
  .then(() => {
    concurrency--;
    drainQueue();
    finished++;
    if (finished === count) {
      resolve();
    }
  });
};
const drainQueue = () => {
  while (params.length > 0 && concurrency < limit) {
    var param = params.shift();
    run(param);
  }
};
drainQueue();
});
uploadFile('localfile');
```

说明

大文件分段上传时，使用**Offset**参数和**PartSize**参数配合指定每段数据在文件中的起始结束位置。

注意

并发数过大，可能会因为网络不稳定等原因，产生Timeout错误，需要限制并发数。

9.2 初始化多段上传任务(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

使用多段上传特性时，用户必须首先调用创建多段上传任务接口创建任务，系统会给用户返回一个全局唯一的多段上传任务号，作为任务标识。后续用户可以根据这个标识发起相关的请求，如：上传段、合并段、列举段等。创建多段上传任务不影响已有的同名对象；同一个对象可以同时存在多个多段上传任务；每个多段上传任务在初始化时可以附加消息头信息，包括acl、用户自定义元数据和通用的HTTP消息头contentType、contentEncoding等，这些附加的消息头信息将先记录在多段上传任务元数据中。调用初始化分段上传任务接口成功后，会返回分段上传任务的全局唯一标识（Upload ID），在后面的操作中将它用到。

接口约束

- 您必须是桶所有者或拥有初始化分段上传任务的权限，才能初始化分段上传任务。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObject权限，如果使用桶策略则需授予PutObject权限。相关授权

方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。

- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 初始化上传段任务并上传一个或多个段之后，您必须合并段或取消多段上传任务，才能停止收取已上传的段的存储费用。仅当在合并段或取消多段上传任务之后，OBS才释放段存储并停止向您收取段存储费用。

方法定义

ObsClient.initiateMultipartUpload(params)

请求参数

表 9-1 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。• 桶命名规则如下：<ul style="list-style-type: none">- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。- 禁止使用IP地址。- 禁止以“-”或“.”开头及结尾。- 禁止两个“.”相邻（如：“my..bucket”）。- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。• 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
ACL	AclType	可选	<p>参数解释: 创桶时可指定的预定义访问策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见ACL功能介绍。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的访问策略选项参见AclType。</p> <p>默认取值: 私有权限</p>

参数名称	参数类型	是否必选	描述
WebsiteRedirectLocation	string	可选	<p>参数解释: 当桶设置了Website配置, 可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。 例如, 重定向请求到桶内另一对象: WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL: WebsiteRedirectLocation:http://www.example.com/ OBS将这个值从头域中取出, 保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>取值范围: 无</p> <p>约束限制:</p> <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向, 不支持为桶中文件夹下的对象设置重定向。 <p>默认取值: 无</p>
ContentType	string	可选	<p>参数解释: 待上传对象的文件类型 (MIME类型)。Content-Type (MIME) 用于标识发送或接收数据的类型, 浏览器根据该参数来决定数据的打开方式。</p> <p>约束限制: 无</p> <p>取值范围: 常见的Content-Type (MIME) 列表参见如何理解Content-Type (MIME) (Node.js SDK)。</p> <p>默认取值: 如果您在上传Object时未指定Content-Type, SDK会根据指定Object名称的后缀名来判定文件类型并自动填充Content-Type。</p>

参数名称	参数类型	是否必选	描述
StorageClass	StorageClassType	可选	<p>参数解释: 对象的存储类别。如果未设置，则为桶的存储类别。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的访问策略选项参见 StorageClassType。</p> <p>默认取值: STANDARD，即标准存储类别</p>
Metadata	object	可选	<p>参数解释: 待上传对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据，以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时，加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为：每个键和值的UTF-8 编码中的字节总数。• 自定义元数据的key值不区分大小写，OBS统一转为小写进行存储。value值区分大小写。• 自定义元数据key-value对都必须符合US-ASCII。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。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantRead	string	可选	<p>参数解释: 被授予READ权限的账号的账号ID, 即 domain_id。账号被授予READ权限后, 针对当前对象, 允许被授权指定账号有读对象和获取对象元数据的权限。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantReadAcp	string	可选	<p>参数解释: 被授予READ_ACP权限的账号的账号ID, 即domain_id。账号被授予READ_ACP权限后, 针对当前对象, 允许读对象的ACL。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantWriteAcp	string	可选	<p>参数解释: 被授予WRITE_ACP权限的账号的账号ID, 即domain_id。账号被授予WRITE_ACP权限后, 针对当前对象, 允许修改对象的ACL信息。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantFullControl	string	可选	<p>参数解释: 被授予FULL_CONTROL权限的账号的账号ID, 即domain_id。账号被授予FULL_CONTROL权限后, 针对当前对象, 授权指定账号有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
Expires	number	可选	<p>参数解释: 表示对象的过期时间(从对象最后修改时间开始计算), 过期之后对象会被自动删除。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 设置的天数计算出的过期时间不能早于当前时间, 如10天前上传的对象, 不能设置小于10的值。 • 此字段对于每个对象仅支持上传时配置, 不支持后期通过修改元数据接口修改。 <p>取值范围: 1~(2⁶³-1), 单位: 天。</p> <p>默认取值: 无</p>
SseKms	string	当使用SSE-KMS方式时, 必选	<p>参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p>约束限制: 当前仅支持kms。</p> <p>取值范围: 可选值: kms, 即选择SSE-KMS方式加密对象。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SseKmsKey	string	可选	<p>参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p>约束限制: 有效值支持两种格式: 1. regionID:domainID(账号ID):key/key_id 2. key_id</p> <p>其中:</p> <ul style="list-style-type: none">• regionID是使用密钥所属region的ID,可在地区和终端节点页面获取;• domainID是使用密钥所属账号的账号ID,获取方法参见如何获取账号ID和用户ID? (Node.js SDK);• key_id是从数据加密服务创建的密钥ID,获取方法请参见查看密钥。 <p>取值范围: 无</p> <p>默认取值:</p> <ul style="list-style-type: none">• 如果用户没有提供该头域,那么默认的主密钥将会被使用。• 如果默认主密钥不存在,将默认创建并使用。
SseC	string	当使用SSE-C方式时,必选	<p>参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p> <p>约束限制: 仅支持AES256。</p> <p>取值范围: 可选值: AES256, 即选择SSE-C方式加密对象。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SseCKey	string	当使用SSE-C方式时，必选	<p>参数解释： SSE-C方式下加密使用的密钥，该密钥用于加密对象。</p> <p>约束限制： 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw=</p> <p>取值范围： 无</p> <p>默认取值： 无</p>
EncodingType	string	可选	<p>参数解释： 用于指定对响应中的Key进行指定类型的编码。如果Key包含xml 1.0标准不支持的控制字符，可通过设置该参数对响应中的Key进行编码。</p> <p>取值范围： 可选值为url。</p> <p>默认取值： 无，不设置则不编码。</p>

表 9-2 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限，其他任何人都没有访问权限。
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上，所有人可以获取该对象内容和元数据。

常量名	原始值	说明
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	桶公共读，桶内对象公共读。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadDelivered不能应用于对象。
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	桶公共读写，桶内对象公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadWriteDelivered不能应用于对象。

常量名	原始值	说明
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	<p>设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。</p> <p>默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。</p>

表 9-3 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。</p>
ObsClient.enums.StorageClassWarm	WARM	<p>低频访问存储。</p> <p>低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。</p>
ObsClient.enums.StorageClassCold	COLD	<p>归档存储。</p> <p>归档存储适用于很少访问（平均一年访问一次）数据的业务场景。</p>

返回结果

表 9-4 返回结果

参数类型	描述
<p>表9-5</p> <p>说明</p> <p>该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释：</p> <p>接口返回信息，详见表 9-5。</p>

表 9-5 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表9-7	参数解释: 操作成功后的结果数据，详见 表9-7 。 约束限制: 当Status大于300时为空。

表 9-6 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 9-7 InitiateMultipartUploadOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
Bucket	string	参数解释: 分段上传任务的桶名。
Key	string	参数解释: 分段上传任务的对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。
UploadId	string	参数解释: 分段上传任务的ID，例如： 000001648453845DBB78F2340DD460D8。
SseKms	string	参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。
SseKmsKey	string	参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。
SseC	string	参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。
SseCKeyMd5	string	参数解释: SSE-C方式下加密使用密钥的MD5值，该值用于验证密钥传输过程中是否出错。 约束限制: 由密钥值经过MD5加密再经过Base64编码后得到，示例：4XvB3tbNTN+tIEVa0/fGaQ==

参数名称	参数类型	描述
EncodingType	string	参数解释: 用于指定对响应中的Key进行指定类型的编码。如果Key包含xml 1.0标准不支持的控制字符, 可通过设置该参数对响应中的Key进行编码。 取值范围: 可选值为url。 默认取值: 无, 不设置则不编码。

代码示例

本示例用于初始化examplebucket桶下的上传任务, 并且指定上传对象名为example/objectname。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function initiateMultipartUpload() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名, 此处以 example/objectname 为例
      Key: "example/objectname",
    };
    // 初始化上传段任务
    const result = await obsClient.initiateMultipartUpload(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Initiate multipart upload successfull with bucket(%s) and object(%s)!", params.Bucket, params.Key);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      // 分段上传任务的全局唯一标识, 在后面的上传段和合并段操作中将它用到它。
      console.log("UploadId: %s", result.InterfaceResult.UploadId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.error(error);
  }
}
```

```
    } catch (error) {  
      console.log("An Exception was found, which means the client encountered an internal problem when  
      attempting to communicate with OBS, for example, the client was unable to access the network.");  
      console.log(error);  
    };  
  };  
};  
  
initiateMultipartUpload();
```

相关链接

- 关于分段上传-上传段的API说明，请参见[上传段](#)。
- 分段上传过程中返回的错误码含义、问题原因及处理措施可参考[OBS错误码](#)。

9.3 上传段(Node.js SDK)

功能说明

初始化分段上传任务后，通过分段上传任务的ID，上传段到指定桶中。

上传段时，除了指定上传ID，还必须指定段编号。您可以选择1和10000之间的任意段编号。段编号在您正在上传的对象中唯一地标示了段及其位置。如果您使用之前上传的段的同一段编号上传新段，则之前上传的段将被覆盖。无论您何时上传段，OBS都将在其响应中返回ETag标头。对于每个段上传任务，您必须记录每个段编号和ETag值。您在后续的合并请求中需要添加这些值以完成多段上传。

接口约束

- 您必须是桶拥有者或拥有合并段的权限，才能合并段。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObject权限，如果使用桶策略则需授予PutObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 上传段接口要求除最后一段以外，其他段的大小范围是100KB~5GB，最后一段的大小范围是0~5GB。但是上传段接口并不会立即校验上传段的大小（因为不知道是否为最后一块）；只有调用合并段接口时才会校验。
- 上传的段的编号有范围限制，其范围是1~10000。
- OBS 3.0的桶支持最小段的大小为100KB，OBS 2.0的桶支持最小段的大小为5MB。请在OBS 3.0的桶上执行分段上传操作。

方法定义

```
ObsClient.uploadPart(params)
```


请求参数

表 9-8 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
PartNumber	number	必选	<p>参数解释: 段号。</p> <p>约束限制: 无</p> <p>取值范围: 取值范围是[1,10000]的非负整数。</p> <p>默认取值: 无</p>
UploadId	string	必选	<p>参数解释: 分段上传任务的ID。任务ID可以通过初始化分段上传任务生成。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>例如： 000001648453845DBB78F2340DD460D8。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
ContentMD5	string	可选	<p>参数解释: 待上传对象数据的MD5值（经过Base64编码），提供给OBS服务端，校验数据完整性。</p> <p>约束限制: 消息体128-bit MD5值经过base64编码后得到的字符串。</p> <p>取值范围: 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。</p> <p>示例：n58IG6hfM7vqI4K0vnWpog==。</p> <p>默认取值: 无</p>
Body	string stream.Readable	可选	<p>参数解释: 待上传对象的数据流。</p> <p>约束限制: 单次上传大小范围是[0, 5GB]。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
SourceFile	string	可选	<p>参数解释: 待上传对象的源文件路径。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 单次上传大小范围是[0, 5GB]。• Body与SourceFile不能同时使用。• 当Body与SourceFile都为空时，上传对象的大小为0字节。• Offset、PartSize和SourceFile配套使用，用于指定上传源文件中的某一段数据。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Offset	number	可选	<p>参数解释: 源文件中某一分段的起始偏移大小。</p> <p>约束限制: Offset、PartSize和SourceFile配套使用,用于指定上传源文件中的某一分段数据。</p> <p>取值范围: 非负整数,小于待上传对象的大小,单位:字节。</p> <p>默认取值: 0</p>
PartSize	number	可选	<p>参数解释: 当前段的长度。</p> <p>约束限制: Offset、PartSize和SourceFile配套使用,用于指定上传源文件中的某一分段数据。</p> <p>取值范围: 100KB~5GB,单位:字节。</p> <p>默认取值: 102400字节</p>
SseKms	string	当使用SSE-KMS方式时,必选	<p>参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p>约束限制: 当前仅支持kms。</p> <p>取值范围: 可选值: kms,即选择SSE-KMS方式加密对象。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SseKmsKey	string	可选	<p>参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p>约束限制: 有效值支持两种格式:</p> <ol style="list-style-type: none">1. regionID:domainID(账号ID):key/key_id2. key_id <p>其中:</p> <ul style="list-style-type: none">• regionID是使用密钥所属region的ID,可在地区和终端节点页面获取;• domainID是使用密钥所属账号的账号ID,获取方法参见如何获取账号ID和用户ID? (Node.js SDK);• key_id是从数据加密服务创建的密钥ID,获取方法请参见查看密钥。 <p>取值范围: 无</p> <p>默认取值:</p> <ul style="list-style-type: none">• 如果用户没有提供该头域,那么默认的主密钥将会被使用。• 如果默认主密钥不存在,将默认创建并使用。
SseC	string	当使用SSE-C方式时,必选	<p>参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p> <p>约束限制: 仅支持AES256。</p> <p>取值范围: 可选值: AES256, 即选择SSE-C方式加密对象。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SseCKey	string	当使用SSE-C方式时，必选	<p>参数解释： SSE-C方式下加密使用的密钥，该密钥用于加密对象。</p> <p>约束限制： 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNvaobncnLht/rCB2o/9Cw=</p> <p>取值范围： 无</p> <p>默认取值： 无</p>

返回结果

表 9-9 返回结果

参数类型	描述
<p>表9-10</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表9-10。</p>

表 9-10 Response

参数名称	参数类型	描述
CommonMsg	ICCommonMsg	<p>参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICCommonMsg。</p>
InterfaceResult	表9-12	<p>参数解释： 操作成功后的结果数据，详见表9-12。</p> <p>约束限制： 当Status大于300时为空。</p>

表 9-11 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 9-12 UploadPartOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
ETag	string	参数解释: 对象的base64编码的128位MD5摘要。ETag是对象内容的唯一标识，可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A，下载对象时ETag为B，则说明对象内容发生了变化。ETag只反映变化的内容，而不是其元数据。上传的对象或复制操作创建的对象，都有唯一的ETag。 约束限制: 当对象是服务端加密的对象时，ETag值不是对象的MD5值。

代码示例

本示例用于通过指定多段上传任务号（uploadId），上传段到examplebucket桶中，并且指定对象名为example/objectname。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function uploadPart() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定分段上传任务的ID
      UploadId: "0000000xxxxxxx",
      // 指定上传段的段号，取值为从1到10000的整数。这次以1为例
      PartNumber: 1,
      // 指定待上传段的源文件路径，这里以localfile为例
      SourceFile: "localfile",
      // 指定分段大小，这里以5MB为例，单位是字节
      PartSize: 5 * 1024 * 1024,
      // 指定该分段的起始偏移大小，这里以0为例
      Offset: 0
    };
    // 上传段
    const result = await obsClient.uploadPart(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Upload part(%d) successful with bucket(%s) and object(%s)!", params.PartNumber,
        params.Bucket, params.Key);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("ETag: %s", result.InterfaceResult.ETag);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

uploadPart();
```


9.4 合并段(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

如果用户上传完所有的段，就可以调用合并段接口，系统将在服务端将用户指定的段合并成一个完整的对象。在执行“合并段”操作以前，用户不能下载已经上传的数据。在合并段时需要将多段上传任务初始化时记录的附加消息头信息复制到对象元数据中，其处理过程和普通上传对象带这些消息头的处理过程相同。在并发合并段的情况下，仍然遵循Last Write Win策略，但“Last Write”的时间定义为段任务的初始化时间。

已经上传的段，只要没有取消对应的多段上传任务，都要占用用户的容量配额；对应的多段上传任务“合并段”操作完成后，只有指定的多段数据占用容量配额，用户上传的其他此多段任务对应的段数据如果没有包含在“合并段”操作指定的段列表中，“合并段”完成后系统将删除多余的段数据，且同时释放容量配额。

合并段时，OBS通过按升序的段编号规范化多段来创建对象。如果在初始化上传段任务中提供了任何对象元数据，则OBS会将该元数据与对象相关联。成功完成请求后，段将不再存在。合并段请求必须包括上传ID以及段编号和相应的ETag值的列表。OBS响应包括可唯一地识别组合对象数据的ETag。此ETag无需成为对象数据的MD5哈希。

接口约束

- 您必须是桶拥有者或拥有合并段的权限，才能合并段。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObject权限，如果使用桶策略则需授予PutObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 如果上传了10个段，但合并时只选择了9个段进行合并，那么未被合并的段将会被系统自动删除，未被合并的段删除后不能恢复。在进行合并之前请使用列出已上传的段接口进行查询，仔细核对所有段，确保没有段被遗漏。
- 如果最后一个段之外的其它段尺寸过小（小于100KB），OBS返回400 Bad Request。

方法定义

```
ObsClient.completeMultipartUpload(params)
```

请求参数

表 9-13 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
UploadId	string	必选	<p>参数解释: 分段上传任务的ID。任务ID可以通过初始化分段上传任务生成。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串，例如： 000001648453845DBB78F2340DD460D8。</p> <p>默认取值: 无</p>
Parts	Part []	必选	<p>参数解释: 待合并的段列表。详情参见Part。</p>
EncodingType	string	可选	<p>参数解释: 对响应中的Key进行指定类型的编码。如果Key包含xml 1.0标准不支持的控制字符，可通过设置该参数对响应中的Key进行编码。</p> <p>取值范围: 可选值为url。</p> <p>默认取值: 无，不设置则不编码。</p>

表 9-14 Part

参数名称	参数类型	描述
PartNumber	number	<p>参数解释: 段号。</p> <p>约束限制: 无</p> <p>取值范围: 取值范围是[1,10000]的非负整数。</p> <p>默认取值: 无</p>

参数名称	参数类型	描述
Etag	string	<p>参数解释: 段的Etag值。分段的base64编码的128位MD5摘要。</p> <p>约束限制: 通过上传段请求可以收到服务端返回的Etag值（段数据的MD5值）返回给用户。</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>

返回结果

表 9-15 返回结果

参数类型	描述
<p>表9-16 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表9-16。</p>

表 9-16 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	<p>参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICommonMsg。</p>
InterfaceResult	表9-18	<p>参数解释: 操作成功后的结果数据，详见表9-18。</p> <p>约束限制: 当Status大于300时为空。</p>

表 9-17 ICommonMsg

参数名称	参数类型	描述
------	------	----

Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 9-18 CompleteMultipartUploadOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
ETag	string	参数解释: 合并段后根据各个段的ETag值计算出的结果。对象的base64编码的128位MD5摘要。ETag是对象内容的唯一标识，可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A，下载对象时ETag为B，则说明对象内容发生了变化。ETag只反映变化的内容，而不是其元数据。上传的对象或复制操作创建的对象，都有唯一的ETag。 约束限制: 当对象是服务端加密的对象时，ETag值不是对象的MD5值。
Bucket	string	参数解释: 合并段所在的桶名。

参数名称	参数类型	描述
Key	string	参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。
Location	string	参数解释: 合并段后得到的对象的url。 例如: https://example-Bucket.obs.regions.myhuaweicloud.com/example-Object
VersionId	string	参数解释: 合并段后得到的对象版本号。如果桶的多版本状态为开启，则会返回对象的版本号。
SseKms	string	参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。
SseKmsKey	string	参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。
SseC	string	参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。
SseCKeyMd5	string	参数解释: SSE-C方式下加密使用密钥的MD5值，该值用于验证密钥传输过程中是否出错。 约束限制: 由密钥值经过MD5加密再经过Base64编码后得到，示例: 4XvB3tbNTN+tIEVa0/fGaQ==
EncodingType	string	参数解释: 用于指定对响应中的Key进行指定类型的编码。如果Key包含xml 1.0标准不支持的控制字符，可通过设置该参数对响应中的Key进行编码。 取值范围: 可选值为url。 默认取值: 无，不设置则不编码。

代码示例

本示例用于合并指定桶 examplebucket 中已上传的段，并且对象名是example/objectname，上传任务ID是 00000188677110424014075CC4A77xxx。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function completeMultipartUpload() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定多段上传任务号，此处以 00000188677110424014075CC4A77xxx 为例
      Uploadid: "00000188677110424014075CC4A77xxx",
      // 指定待合并的段列表，该列表必须按分段号升序排列，且分段可以是不连续的
      Parts: [
        { PartNumber: 1, ETag: "etag1" },
        { PartNumber: 2, ETag: "etag2" },
        { PartNumber: 3, ETag: "etag3" },
      ]
    };
    // 合并段
    const result = await obsClient.completeMultipartUpload(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Complete multipart upload successful with bucket(%s) and object(%s)!", params.Bucket, params.Key);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

completeMultipartUpload();
```

9.5 列举已上传的段(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

通过分段上传任务的ID，列举指定桶中已上传的段。

您可以列出特定多段上传任务或所有正在进行的多段上传任务的分段。列举已上传的段操作将返回您已为特定多段上传任务而上传的段信息。对于每个列举已上传的段请求，OBS将返回有关特定多段上传任务的分段信息，最多为1000个分段。如果多段上传中的段超过1000个，您必须发送一系列列举已上传的段请求以检索所有段。请注意，返回的分段列表不包括已合并的分段。

接口约束

- 您必须是桶拥有者或拥有列举已上传的段的权限，才能列举已上传的段。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:ListMultipartUploadParts权限，如果使用桶策略则需授予ListMultipartUploadParts权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 列举段至多返回1000个段信息，如果指定的Upload ID包含的段数量大于1000，则返回结果中InterfaceResult.IsTruncated为true表明本次没有返回全部段，并可通过InterfaceResult.NextPartNumberMarker获取下次列举的起始位置。

方法定义

```
ObsClient.listParts(params)
```


请求参数

表 9-19 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
UploadId	string	必选	<p>参数解释: 分段上传任务的ID。任务ID可以通过初始化分段上传任务生成。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串，例如： 000001648453845DBB78F2340DD460D8。</p> <p>默认取值: 无</p>
PartNumberMarker	number	可选	<p>参数解释: 列举已上传段的起始位置。</p> <p>约束限制: 无</p> <p>约束限制: 只有PartNumber数目大于该参数的Part会被列出</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
MaxParts	number	可选	参数解释: 列举已上传段的返回结果最大段数目, 即分页时每一页中段数目。 约束限制: 如果该参数超出1000时, 则按照默认的1000进行处理。 取值范围: 范围是[1, 1000] 默认取值: 1000
EncodingType	string	可选	参数解释: 用于指定对响应中的Key进行指定类型的编码。如果Key包含xml 1.0标准不支持的控制字符, 可通过设置该参数对响应中的Key进行编码。 约束限制: 无 取值范围: 可选值为url。 默认取值: 无, 不设置则不编码。

返回结果

表 9-20 返回结果

参数类型	描述
表9-21 说明 该接口返回是一个Promise类型, 需要使用Promise、async/await语法处理。	参数解释: 接口返回信息, 详见表 9-21。

表 9-21 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表9-23	参数解释: 操作成功后的结果数据，详见 表9-23 。 约束限制: 当Status大于300时为空。

表 9-22 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 9-23 ListMultipartUploadsOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
Bucket	string	参数解释: 桶名。
Key	string	参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您的对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。
UploadId	string	参数解释: 分段上传任务的ID，例如： 000001648453845DBB78F2340DD460D8
Initiator	Initiator	参数解释: 分段上传任务的创建者，详见 Initiator 。
Owner	Owner	参数解释: 和initiator相同，代表分段上传任务的创建者，详见 Owner 。
StorageClass	StorageClass Type	参数解释: 待合并对象的存储类别。存储类别取值参见 StorageClassType 。
PartNumberMarker	number	参数解释: 列举段的起始位置，与请求中的该参数对应。
NextPartNumberMarker	number	参数解释: 下次列举段请求的起始位置。如果本次没有返回全部结果，响应请求中将包含此字段，用于标明接下来请求的PartNumberMarker值。
MaxParts	number	参数解释: 列举段的返回结果最大段数目，与请求中的该参数对应。列举已上传段的返回结果最大段数目，即分页时每一页中段数目。

参数名称	参数类型	描述
IsTruncated	boolean	参数解释: 表明本次请求是否返回了全部结果。 取值范围: <ul style="list-style-type: none"> • true: 表示没有返回全部结果。 • false: 表示已返回了全部结果。
Parts	Part []	参数解释: 已上传段信息列表, 详见 Part 。
EncodingType	string	参数解释: 用于指定对响应中的Key进行指定类型的编码。如果Key包含xml 1.0标准不支持的控制字符, 可通过设置该参数对响应中的Key进行编码。

表 9-24 Initiator

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时必选	参数解释: 创建者的账号ID, 即domain_id。 取值范围: 如何获取账号ID请参见 如何获取账号ID和用户ID? (Node.js SDK) 默认取值: 无
Display Name	string	可选	参数解释: 创建者账号名。 约束限制: 账号名只能包含英文字母、数字或特殊字符“- _”, 且须以字母开头, 账号名长度支持6~32个字符。 默认取值: 无

表 9-25 Owner

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时必选	<p>参数解释: 所有者的账号ID, 即domain_id。</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
Display Name	string	可选	<p>参数解释: 所有者的账号用户名。</p> <p>默认取值: 无</p>

表 9-26 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量, 适用于有大量热点对象(平均一个月多次)或小对象(<1MB), 且需要频繁访问数据的业务场景。</p>
ObsClient.enums.StorageClassWarm	WARM	<p>低频访问存储。</p> <p>低频访问存储适用于不频繁访问(平均一年少于12次)但在需要时也要求能够快速访问数据的业务场景。</p>
ObsClient.enums.StorageClassCold	COLD	<p>归档存储。</p> <p>归档存储适用于很少访问(平均一年访问一次)数据的业务场景。</p>

表 9-27 Part

参数名称	参数类型	描述
PartNumber	number	<p>参数解释: 段号。</p>
LastModified	string	<p>参数解释: 目标段的最近一次修改时间(UTC时间)。</p>

参数名称	参数类型	描述
ETag	string	参数解释: 段的ETag值。分段的base64编码的128位MD5摘要。
Size	number	参数解释: 段占用的空间大小。

代码示例

本示例用于查询一个任务所属的所有段信息。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');
// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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
  // 您可以通过登录访问管理控制台获取访问密钥AK/SK, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listParts() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名, 此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定多段上传任务号, 此处以 00000188677110424014075CC4A77xxx 为例
      UploadId: "00000188677110424014075CC4A77xxx",
      // 指定返回的最大对象数, 此处以 100 为例, 返回的对象列表将是按照字典顺序的最多前max-keys个对象, 默认值为1000
      MaxKeys: 100,
    };
    // 列举已上传段
    const result = await obsClient.listParts(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List part successful with bucket(%s) and object(%s)!", params.Bucket, params.Key);
      console.log('RequestId: %s', result.CommonMsg.RequestId);
      console.log('Bucket: %s', result.InterfaceResult.Bucket);
      console.log('Key: %s', result.InterfaceResult.Key);
      console.log('UploadId: %s', result.InterfaceResult.UploadId);
      console.log('PartNumberMarker: %s', result.InterfaceResult.PartNumberMarker);
      console.log('NextPartNumberMarker: %s', result.InterfaceResult.NextPartNumberMarker);
      console.log('MaxParts: %s', result.InterfaceResult.MaxParts);
      console.log('IsTruncated: %s', result.InterfaceResult.IsTruncated);
      console.log('StorageClass: %s', result.InterfaceResult.StorageClass);
      console.log('Initiator[ID]: %s', result.InterfaceResult.Initiator['ID']);
      console.log('Owner[ID]: %s', result.InterfaceResult.Owner['ID']);
      for (let i = 0; i < result.InterfaceResult.Parts.length; i++) {
        const part = result.InterfaceResult.Parts[i];
        console.log("Part[%d]-ETag:%s, PartNumber:%d, LastModified:%s, Size:%d",

```



```
        i, part.PartNumber, part.LastModified, part.ETag, part.Size);
    };
    return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an
error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
};
};
listParts();
```

9.6 列举分段上传任务(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

列举指定桶中所有的初始化后还未合并或还未取消的分段上传任务。

通过列举桶中的多段上传任务，您可以获得已初始化多段上传任务的列表，已初始化多段上传任务是指初始化后还未合并以及未取消的多段上传任务。每个请求将返回最多1000个多段上传任务，如果指定的桶包含的分段上传任务数量大于1000，则返回结果中InterfaceResult.IsTruncated为true表明本次没有返回全部结果，并可通过InterfaceResult.NextKeyMarker和InterfaceResult.NextUploadIdMarker获取下次列举的起点。

接口约束

- 您必须是桶拥有者或拥有列举分段上传任务的权限，才能列举分段上传任务。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:ListBucketMultipartUploads权限，如果使用桶策略则需授予ListBucketMultipartUploads权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.listMultipartUploads(params)
```

请求参数

表 9-28 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my.bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
Prefix	string	可选	<p>参数解释: 限定返回的分段上传任务中的对象名必须带有prefix前缀。</p> <p>例如, 假设您拥有以下对象: logs/day1、logs/day2、logs/day3和ExampleObject.jpg。如果您将logs/指定为前缀, 将返回以字符串“logs/”开头的三个对象所在的分段上传任务。如果您指定空的前缀且请求中没有其他过滤条件, 将返回桶中的所有分段上传任务。</p> <p>约束限制: 长度大于0且不超过1024的字符串。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
MaxUploads	number	可选	<p>参数解释: 列举分段上传任务的最大数目。</p> <p>约束限制: 当该参数超出1000时, 按照默认的1000进行处理。</p> <p>取值范围: 1~1000, 单位: 个。</p> <p>默认取值: 1000</p>

参数名称	参数类型	是否必选	描述
Delimiter	string	可选	<p>参数解释: 对分段上传任务中的对象名进行分组的字符。通常与前缀prefix搭配使用, 如果指定了prefix, 从prefix到第一次出现delimiter间具有相同字符串的对象名会被分成一组, 形成一条CommonPrefixes; 如果没有指定prefix, 从对象名的首字符到第一次出现delimiter间具有相同字符串的对象名会被分成一组, 形成一条CommonPrefixes。</p> <p>例如, 桶中有3个对象, 分别为abcd、abcde、bbcde。如果指定delimiter为d, prefix为a, abcd、abcde会被分成一组, 形成一条前缀为abcd的CommonPrefixes; 如果只指定delimiter为d, abcd、abcde会被分成一组, 形成一条前缀为abcd的CommonPrefixes, 而bbcde会被单独分成一组, 形成一条前缀为bbcd的CommonPrefixes。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
KeyMarker	string	可选	<p>参数解释: 列举分段上传任务的起始位置。表示列举时返回指定的KeyMarker之后的分段上传任务。</p> <p>约束限制: 无</p> <p>取值范围: 上次请求返回体的NextKeyMarker值。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
UploadIdMarker	string	可选	参数解释: 列举分段上传任务的起始位置（uploadId标识）。 约束限制: 只有与KeyMarker参数一起使用时才有意义，即列举时返回指定KeyMarker的UploadIdMarker之后的分段上传任务。 取值范围: 对象的分段上传任务ID，即上次请求返回体的NextUploadIdMarker值。 默认取值: 无
EncodingType	string	可选	参数解释: 用于指定对响应中的Key进行指定类型的编码。如果Key包含xml 1.0标准不支持的控制字符，可通过设置该参数对响应中的Key进行编码。 约束限制: 无 取值范围: 可选值为url。 默认取值: 无，不设置则不编码。

返回结果

表 9-29 返回结果

参数类型	描述
表9-30 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释: 接口返回信息，详见表9-30。

表 9-30 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表9-32	参数解释: 操作成功后的结果数据，详见 表9-32 。 约束限制: 当Status大于300时为空。

表 9-31 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 9-32 ListMultipartUploadsOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
Bucket	string	参数解释: 桶名。
KeyMarker	string	参数解释: 列举分段上传任务的起始位置，与请求中的该参数对应。表示列举时返回指定的KeyMarker之后的分段上传任务。
UploadIdMarker	string	参数解释: 列举分段上传任务的起始位置（uploadId标识），与请求中的该参数对应。
NextKeyMarker	string	参数解释: 下次列举分段上传任务请求的起始位置。如果本次没有返回全部结果，响应请求中将包含该元素，用于标明接下来请求的KeyMarker值。
NextUploadIdMarker	string	参数解释: 下次列举分段上传任务请求的起始位置（uploadId标识）。与NextKeyMarker配合使用。如果本次没有返回全部结果，响应请求中将包含该元素，用于标明接下来请求的UploadIdMarker值。
MaxUploads	number	参数解释: 列举分段上传任务的最大数目，与请求中的该参数对应。
IsTruncated	boolean	参数解释: 表明本次请求是否返回了全部结果。 取值范围: <ul style="list-style-type: none">• true: 表示没有返回全部结果。• false: 表示已返回了全部结果。

参数名称	参数类型	描述
Prefix	string	参数解释: 分段上传任务中的对象名前缀，与请求中的该参数对应。限定返回的分段上传任务中的对象名必须带有prefix前缀。 例如，假设您拥有以下对象：logs/day1、logs/day2、logs/day3和ExampleObject.jpg。如果您将logs/指定为前缀，将返回以字符串“logs/”开头的三个对象所在的分段上传任务。如果您指定空的前缀且请求中没有其他过滤条件，将返回桶中的所有分段上传任务。
Delimiter	string	参数解释: 用于对分段上传任务中的对象名进行分组的字符，与请求中的该参数对应。通常与前缀prefix搭配使用，如果指定了prefix，从prefix到第一次出现delimiter间具有相同字符串的对象名会被分成一组，形成一条CommonPrefixes；如果没有指定prefix，从对象名的首字符到第一次出现delimiter间具有相同字符串的对象名会被分成一组，形成一条CommonPrefixes。 例如，桶中有3个对象，分别为abcd、abcde、bbcde。如果指定delimiter为d，prefix为a，abcd、abcde会被分成一组，形成一条前缀为abcd的CommonPrefixes；如果只指定delimiter为d，abcd、abcde会被分成一组，形成一条前缀为abcd的CommonPrefixes，而bbcde会被单独分成一组，形成一条前缀为bbcd的CommonPrefixes。
Uploads	Upload []	参数解释: 桶内分段上传任务列表，详见 Upload 。
EncodingType	string	参数解释: 用于指定对响应中的Key进行指定类型的编码。如果Key包含xml 1.0标准不支持的控制字符，可通过设置该参数对响应中的Key进行编码。

表 9-33 Upload

参数名称	参数类型	描述
Key	string	参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。
UploadId	string	参数解释: 分段上传任务的ID，例如： 000001648453845DBB78F2340DD460D8
Initiator	Initiator	参数解释: 分段上传任务的创建者，详见 Initiator 。
Owner	Owner	参数解释: 分段上传的对象的所有者账号ID，详见 Owner 。
StorageClass	StorageClassType	参数解释: 分段上传的对象的存储类别，存储类别取值范围参见 StorageClassType 。
Initiated	string	参数解释: 分段上传任务的初始化时间。

表 9-34 Initiator

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时必选	参数解释: 创建者的账号ID，即 domain_id。 取值范围: 如何获取账号ID请参见 如何获取账号ID和用户ID? (Node.js SDK) 默认取值: 无

参数名称	参数类型	是否必选	描述
Display Name	string	可选	<p>参数解释: 创建者账号名。</p> <p>约束限制: 账号名只能包含英文字母、数字或特殊字符“-_”，且须以字母开头，账号名长度支持6~32个字符。</p> <p>默认取值: 无</p>

表 9-35 Owner

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时必选	<p>参数解释: 所有者的账号ID，即domain_id。</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
Display Name	string	可选	<p>参数解释: 所有者的账号用户名。</p> <p>默认取值: 无</p>

表 9-36 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。</p>
ObsClient.enums.StorageClassWarm	WARM	<p>低频访问存储。</p> <p>低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。</p>

常量名	原始值	说明
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

代码示例

本示例用于列举examplebucket桶中所有的初始化后还未合并以及未取消的多段上传任务。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listMultipartUploads() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例。
      Key: "example/objectname",
      // 指定列举的多段任务的最大条目，此处以10为例。
      MaxUploads: 10
    };
    // 列举分段上传任务
    const result = await obsClient.listMultipartUploads(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List multipart uploads successful with bucket(%s)!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let i = 0; i < output.Uploads.length; i++) {
        const upload = output.Uploads[i];
        console.log("Upload[%d]-OwnerId:%s, UploadId:%s, Key:%s, Initiated:%s",
          i, upload.Owner.ID, upload.UploadId, upload.Key, upload.Initiated);
      }
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}
```

```
};  
listMultipartUploads());
```

9.7 复制段(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

初始化分段上传任务后，通过指定分段上传任务的ID，复制段到指定桶中。

多段上传任务创建后，用户可以通过指定多段上传任务的ID，为特定的任务上传段。添加段的方式还包括调用段复制接口。允许用户将已上传对象的一部分或全部复制为段。

将源对象object复制为一个段part1，如果在复制操作之前part1已经存在，复制操作执行之后，旧的段数据part1会被新复制的段数据覆盖。复制成功后，只能列举到最新的段part1，旧的段数据将会被删除。因此在使用复制段接口时请确保目标段不存在或者已无价值，避免因复制段导致数据误删除。复制过程中源对象object无任何变化。

接口约束

- 您必须是桶拥有者或拥有复制段的权限，才能复制段。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObject权限，如果使用桶策略则需授予PutObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 复制段的结果不能仅根据HTTP返回头域中的status_code来判断请求是否成功，头域中status_code返回200时表示服务端已经收到请求，且开始处理复制段请求。复制是否成功会在响应消息的body中，只有body体中有ETag标签才表示成功，否则表示复制失败。

方法定义

```
ObsClient.copyPart(params)
```

请求参数

表 9-37 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 目标桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
PartNumber	number	必选	<p>参数解释: 段号。</p> <p>约束限制: 无</p> <p>取值范围: 取值范围是[1,10000]。</p> <p>默认取值: 无</p>
UploadId	string	必选	<p>参数解释: 分段上传任务的ID。任务ID可以通过初始化分段上传任务生成。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串，例如： 000001648453845DBB78F2340DD460D8。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
CopySource	string	必选	<p>参数解释: 用来指定复制对象操作的源桶名以及源对象名。当源对象存在多个版本时，通过versionId参数指定版本源对象。</p> <p>示例: source_bucket/sourceObject?versionId=G001117FCE89978B0000401205D5DC9A</p> <p>约束限制: 中文字符与%，需要进行URL编码。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

返回结果

表 9-38 返回结果

参数类型	描述
<p>表9-39</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表9-39。</p>

表 9-39 Response

参数名称	参数类型	描述
CommonMsg	ICCommonMsg	<p>参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICCommonMsg。</p>
InterfaceResult	表9-41	<p>参数解释: 操作成功后的结果数据，详见表9-41。</p> <p>约束限制: 当Status大于300时为空。</p>

表 9-40 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 9-41 CopyPartOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
ETag	string	参数解释: 对象的base64编码的128位MD5摘要。ETag是对象内容的唯一标识，可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A，下载对象时ETag为B，则说明对象内容发生了变化。ETag只反映变化的内容，而不是其元数据。上传的对象或复制操作创建的对象，都有唯一的ETag。 约束限制: 当对象是服务端加密的对象时，ETag值不是对象的MD5值。
LastModified	string	参数解释: 目标段的最近一次修改时间（UTC时间）。

参数名称	参数类型	描述
PartNumber	number	参数解释: 段号。
SseKms	string	参数解释: 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。
SseKmsKey	string	参数解释: SSE-KMS加密方式下使用的KMS主密钥的ID值。
SseC	string	参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。
SseCKeyMd5	string	参数解释: SSE-C方式下加密使用密钥的MD5值，该值用于验证密钥传输过程中是否出错。 约束限制: 由密钥值经过MD5加密再经过Base64编码后得到，示例：4XvB3tbNTN+tIEVa0/fGaQ==

代码示例

本示例用于复制源桶（sourcebucketname）下对象（example/sourceobjectname）的段，到目标桶examplebucket的对象（example/objectname）。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyPart() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定复制源桶名和源对象名，此处复制源桶名以sourcebucketname为例，源对象名以example/sourceobjectname为例
    };
  }
}
```

```
CopySource: "sourcebucketname/example/sourceobjectname",
// 指定上传段的段号, 此处以1为例
PartNumber: 1,
// 指定多段上传任务号, 此处以 00000188677110424014075CC4A77xxx 为例
UploadId: "00000188677110424014075CC4A77xxx",
};
// 复制段
const result = await obsClient.copyPart(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Complete multipart upload successful with bucket(%s) and object(%s)!", params.Bucket,
params.Key);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
copyPart();
```

9.8 取消分段上传任务(Node.js SDK)

须知

开发过程中, 您有任何问题可以在github上[提交issue](#)。

功能说明

通过分段上传任务的ID, 取消指定分段上传任务。

您可以选择取消多段上传任务, 取消多段上传任务之后无法再次使用该上传ID上传任何段。然后, OBS将释放被取消的多段上传任务中的每个段数据的所有存储。如果有多段上传已在进行中, 即使您已执行中止操作, 它们仍可以上传成功或失败。如果要释放所有分段使用的所有存储, 必须在完成所有多段上传后再取消多段上传任务。

接口约束

- 您必须是桶拥有者或拥有取消分段上传任务的权限, 才能取消分段上传任务。建议使用IAM或桶策略进行授权, 如果使用IAM则需授予obs:object:AbortMultipartUpload权限, 如果使用桶策略则需授予AbortMultipartUpload权限。相关授权方式介绍可参见[OBS权限控制概述](#), 配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系, 详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.abortMultipartUpload(params)
```

请求参数

表 9-42 请求参数列表

参数名称	参数类型	是否必选	描述
input	表9-43	是	参数解释: 取消分段上传任务接口入参, 详见表9-43。

表 9-43 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	参数解释: 桶名。 约束限制: <ul style="list-style-type: none">桶的名字需全局唯一, 不能与已有的任何桶名称重复, 包括其他用户创建的桶。桶命名规则如下:<ul style="list-style-type: none">3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻(如: “my..bucket”)。禁止“.”和“-”相邻(如: “my-.bucket”和“my-bucket”)。同一用户在同一个区域多次创建同名桶不会报错, 创建的桶属性以第一次请求为准。 取值范围: 长度为3~63个字符。 默认取值: 无

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
UploadId	string	必选	<p>参数解释: 分段上传任务的ID。任务ID可以通过初始化分段上传任务生成。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串，例如： 000001648453845DBB78F2340DD460D8。</p> <p>默认取值: 无</p>

返回结果

表 9-44 返回结果

参数类型	描述
<p>表9-45</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表 9-45。</p>

表 9-45 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	表9-47	参数解释: 操作成功后的结果数据, 详见 表9-47 。 约束限制: 当Status大于300时为空。

表 9-46 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 9-47 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

本示例用于取消名为examplebucket桶中的分段上传任务，并且对象名是example/objectname，上传任务的ID是00000188677110424014075CC4A77xxx。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function abortMultipartUpload() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定多段上传任务号，此处以 00000188677110424014075CC4A77xxx 为例
      UploadId: "00000188677110424014075CC4A77xxx",
    };
    // 取消分段上传任务
    const result = await obsClient.abortMultipartUpload(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Abort multipart upload successful!");
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

abortMultipartUpload();
```

相关链接

- 关于分段上传-取消分段上传任务的API说明，请参见[取消多段上传任务](#)。
- 分段上传过程中返回的错误码含义、问题原因及处理措施可参考[OBS错误码](#)。

10 临时授权访问(Node.js SDK)

10.1 使用临时 URL 进行授权访问(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

通过访问密钥、请求方法类型、请求参数等信息生成一个在Query参数中携带鉴权信息的URL，可将该URL提供给其他用户进行临时访问。在生成URL时，需要指定URL的有效期来限制访客用户的访问时长。

如果想授予其他用户对桶或对象临时进行其他操作的权限（例如上传或下载对象），则需要生成带对应请求的URL后（例如生成上传对象PUT请求的URL），将该URL提供给其他用户。

接口约束

- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 如果遇到跨域报错、签名不匹配问题，请参考以下步骤排查问题：
 - a. 未配置跨域，需要在控制台配置CORS规则，请参考[配置桶允许跨域请求](#)。
 - b. 签名计算问题，请参考[URL中携带签名](#)排查签名参数是否正确；比如上传对象功能，后端将Content-Type参与计算签名生成授权URL，但是前端使用授权URL时没有设置Content-Type字段或者传入错误的值，此时会出现跨域错误。解决方案为：Content-Type字段前后端保持一致。

方法定义

```
ObsClient.createSignedUrlSync(params)
```


请求参数

表 10-1 请求参数列表

参数名称	参数类型	是否必选	描述
Method	HttpMethodType	必选	参数解释: HTTP方法类型, 详情参见 HttpMethodType 。
Bucket	string	必选	参数解释: 桶名。 约束限制: <ul style="list-style-type: none">桶的名字需全局唯一, 不能与已有的任何桶名称重复, 包括其他用户创建的桶。桶命名规则如下:<ul style="list-style-type: none">3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻(如: “my.bucket”)。禁止“.”和“-”相邻(如: “my-.bucket”和“my-bucket”)。同一用户在同一个区域多次创建同名桶不会报错, 创建的桶属性以第一次请求为准。 取值范围: 长度为3~63个字符。 默认取值: 无

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
SpecialParam	SpecialParam	可选	<p>参数解释: 要访问的子资源。</p> <p>约束限制: 无</p> <p>取值范围: 详情参见SpecialParam。</p> <p>默认取值: 无</p>
Expires	number	可选	<p>参数解释: 带授权信息的URL的过期时间。</p> <p>约束限制: 无</p> <p>取值范围: 0~ (2³¹-1)，单位：秒。</p> <p>默认取值: 300</p>
Headers	object	可选	<p>参数解释: 请求中携带的头域。</p> <p>约束限制: 无</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
QueryParams	object	可选	参数解释: 请求中携带的查询参数。 约束限制: 无 取值范围: 无 默认取值: 无

表 10-2 HttpMethodType

常量值	说明
GET	HTTP GET请求。
POST	HTTP POST请求。
PUT	HTTP PUT请求。
DELETE	HTTP DELETE请求。
HEAD	HTTP HEAD请求。
OPTIONS	HTTP OPTIONS请求。

表 10-3 SpecialParam

常量值	适用接口
storagePolicy	设置/获取桶存储类型。
quota	设置/获取桶配额。
storageinfo	获取桶存量信息。
location	获取桶区域位置。
acl	设置/获取桶ACL、设置/获取对象ACL。
policy	设置/获取/删除桶策略。
cors	设置/获取/删除桶CORS配置。
versioning	设置/获取桶多版本状态。
website	设置/获取/删除桶Website配置。
logging	设置/获取桶日志管理配置。

常量值	适用接口
lifecycle	设置/获取/删除桶生命周期配置。
notification	设置/获取桶时间通知配置。
tagging	设置/获取/删除桶标签。
append	追加上传对象
delete	批量删除对象。
versions	列举桶内多版本对象。
uploads	列举桶内分段上传任务、初始化分段上传任务。
restore	恢复归档存储对象。

返回结果

表 10-4 返回结果

参数名称	参数类型	描述
SignedUrl	string	参数解释: 带授权信息的URL。
ActualSignedRequestHeaders	object	参数解释: 通过带授权信息的URL发起请求时实际应携带的头域。

通过OBS Node.js SDK生成临时URL访问OBS的步骤如下:

步骤1 通过ObsClient.createSignedUrlSync生成带签名信息的URL。

步骤2 使用任意HTTP库发送HTTP/HTTPS请求，访问OBS服务。

----结束

注意

如果遇到跨域报错、签名不匹配问题，请参考以下步骤排查问题：

1. 未配置跨域，需要在控制台配置CORS规则，请参考[配置桶允许跨域请求](#)。
2. 签名计算问题，请参考[URL中携带签名](#)排查签名参数是否正确；比如上传对象功能，后端将Content-Type参与计算签名生成授权URL，但是前端使用授权URL时没有设置Content-Type字段或者传入错误的值，此时会出现跨域错误。解决方案为：Content-Type字段前后端保持一致。

以下代码示例展示了如何使用临时URL进行授权访问，包括：创建桶、上传对象、下载对象、列举对象、删除对象。

代码示例一：创建桶

本示例用于使用临时URL授权创建桶。

```
// 引入obs库
// 使用npm安装
var ObsClient = require('esdk-obs-nodejs');
// 使用源码安装
// var ObsClient = require('./lib/obs');
var https = require('https');
var urlLib = require('url');
var crypto = require('crypto');

// 创建ObsClient实例
var obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  server : 'https://your-endpoint'
});

let bucketName = 'bucketname';
let method = 'PUT';
let res = obsClient.createSignedUrlSync({Method : method, Bucket : bucketName});
let location = 'your-location';
let content = `<CreateBucketConfiguration><LocationConstraint>${location}</LocationConstraint></CreateBucketConfiguration>`;

// 使用PUT请求创建桶
var url = urlLib.parse(res.SignedUrl);
var req = https.request({
  method : method,
  host : url.hostname,
  port : url.port,
  path : url.path,
  rejectUnauthorized : false,
  headers : res.ActualSignedRequestHeaders || {}
});

console.log('Creating bucket using url:' + res.SignedUrl);

req.on('response', (serverback) => {
  var buffers = [];
  serverback.on('data', (data) => {
    buffers.push(data);
  }).on('end', () => {

    if(serverback.statusCode < 300){
      console.log('Creating bucket using temporary signature succeed.');
```

```
if(content){
  req.write(content);
};
req.end();
```

代码示例二：上传对象

本示例用于使用临时URL授权上传对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');
```

```
// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});
```

```
function createSignedUrlSync() {
  // 指定HTTP方法类型，这里以PUT为例
  const method = 'PUT';
  const params = {
    // 指定存储桶名称
    Bucket: "examplebucket",
    // 指定对象名，此处以 example/objectname 为例
    Key: "example/objectname",
    // 指定HTTP方法类型
    Method: method,
    // 指定签名URL的过期时间，这里以3600为例，单位为秒
    Expires: 3600,
    // 指定请求中携带的头域
    Headers: {
      "Content-Type": "text/plain",
    }
  };
};
```

```
// 生成上传对象的带授权信息的URL
const res = obsClient.createSignedUrlSync(params);
console.log("SignedUrl: %s", res.SignedUrl);
console.log("ActualSignedRequestHeaders: %v", res.ActualSignedRequestHeaders);
```

```
let content = 'Hello OBS';
// 使用PUT请求上传对象
var url = urlLib.parse(res.SignedUrl);
var req = https.request({
  method: method,
  host: url.hostname,
  port: url.port,
  path: url.path,
  rejectUnauthorized: false,
  headers: res.ActualSignedRequestHeaders || {}
});
console.log('Creating object using url.' + res.SignedUrl);

req.on('response', (serverback) => {
  var buffers = [];
  serverback.on('data', (data) => {
```

```
    buffers.push(data);
  }).on('end', () => {
    if (serverback.statusCode < 300) {
      console.log('Creating object using temporary signature succeed.');
```

代码示例三：下载对象

本示例用于使用临时URL授权下载对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');
```

```
var req = https.request({
  method: method,
  host: url.hostname,
  port: url.port,
  path: url.path,
  rejectUnauthorized: false,
  headers: res.ActualSignedRequestHeaders || {}
});
console.log('Creating object using url.' + res.SignedUrl);

req.on('response', (serverback) => {
  var buffers = [];
  serverback.on('data', (data) => {
    buffers.push(data);
  }).on('end', () => {
    if (serverback.statusCode < 300) {
      console.log('Getting object using temporary signature succeed.');
```

代码示例四：列举对象

本示例用于使用临时URL授权列举对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

function createSignedUrlSync() {
  // 指定HTTP方法类型，这里以GET为例
  const method = 'GET';

  const params = {
    // 指定存储桶名称
    Bucket: "examplebucket",
```



```
// 指定HTTP方法类型
Method: method,
// 指定签名URL的过期时间, 这里以3600为例, 单位为秒
Expires: 3600,
};

// 使用GET请求获取对象列表
var url = urlLib.parse(res.SignedUrl);
var req = https.request({
  method: method,
  host: url.hostname,
  port: url.port,
  path: url.path,
  rejectUnauthorized: false,
  headers: res.ActualSignedRequestHeaders || {}
});
console.log('Listing object using url:' + res.SignedUrl);

req.on('response', (serverback) => {
  var buffers = [];
  serverback.on('data', (data) => {
    buffers.push(data);
  }).on('end', () => {
    if (serverback.statusCode < 300) {
      console.log('Listing object using temporary signature succeed.');
```

代码示例五：删除对象

本示例用于使用临时URL授权删除对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');
```

```
// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});
```

```
function createSignedUrlSync() {
  // 指定HTTP方法类型, 这里以DELETE为例
  const method = 'DELETE';
  const params = {
    // 指定存储桶名称
    Bucket: "examplebucket",
    // 指定对象名, 此处以 example/objectname 为例
    Key: "example/objectname",
    // 指定HTTP方法类型
    Method: method,
    // 指定签名URL的过期时间, 这里以3600为例, 单位为秒
    Expires: 3600,
  };

  // 使用DELETE请求删除对象
  var url = urlLib.parse(res.SignedUrl);
  var req = https.request({
    method: method,
    host: url.hostname,
    port: url.port,
    path: url.path,
    rejectUnauthorized: false,
    headers: res.ActualSignedRequestHeaders || {}
  });
  console.log('Deleting object using url:' + res.SignedUrl);
  req.on('response', (serverback) => {
    var buffers = [];
    serverback.on('data', (data) => {
      buffers.push(data);
    }).on('end', () => {
      if (serverback.statusCode < 300) {
        console.log('Deleting object using temporary signature succeed!');
      } else {
        console.log('Deleting object using temporary signature failed!');
        console.log('status:' + serverback.statusCode);
        console.log('\n');
      };
      buffers = Buffer.concat(buffers);
      if (buffers.length > 0) {
        console.log(buffers.toString());
      };
      console.log('\n');
    });
  }).on('error', (err) => {
    console.log('Deleting object using temporary signature failed!');
    console.log(err);
    console.log('\n');
  });
  req.end();
};

createSignedUrlSync();
```

11 多版本控制(Node.js SDK)

11.1 多版本控制简介(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

OBS支持保存一个对象的多个版本，您可以利用多版本控制，在一个桶中保留多个版本的对象。

多版本功能可以方便地检索和还原各个版本，在意外操作或应用程序故障时快速恢复数据。

在默认情况下，OBS中新创建的桶不会开启多版本功能，向同一个桶上传同名的对象时，新上传的对象将覆盖原有的对象。

更多关于多版本控制的内容请参见[多版本控制](#)。

11.2 设置桶多版本状态(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

OBS支持多版本控制，您可以在一个桶中保留对象的多个版本，使您更方便地检索和还原各个版本，在意外操作或应用程序故障时快速恢复数据。更多多版本相关信息请参见[多版本控制](#)。

调用设置桶的多版本状态接口，您可以为指定桶设置多版本状态。

接口约束

- 您必须是桶所有者或拥有设置桶的多版本状态的权限，才能设置桶的多版本状态。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutBucketVersioning权限，如果使用桶策略则需授予PutBucketVersioning权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

ObsClient.setBucketVersioning(params)

请求参数

表 11-1 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶• 桶命名规则如下：<ul style="list-style-type: none">- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。- 禁止使用IP地址。- 禁止以“-”或“.”开头及结尾。- 禁止两个“.”相邻（如：“my..bucket”）。- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。• 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
VersionStatus	string	必选	参数解释: 桶的多版本状态。 约束限制: 无 取值范围: 桶的多版本状态, 支持的值: <ul style="list-style-type: none">• Enabled: 启用• Suspended: 暂停 默认取值: 无

返回结果

表 11-2 返回结果

参数类型	描述
表11-3 说明 该接口返回是一个Promise类型, 需要使用Promise、async/await语法处理。	参数解释: 接口返回信息, 详见 表11-3 。

表 11-3 Response

参数名称	参数类型	描述
CommonMsg	ICCommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICCommonMsg 。
InterfaceResult	表11-5	参数解释: 操作成功后的结果数据, 详见 表11-5 。 约束限制: 当Status大于300时为空。

表 11-4 ICommonMsg

参数名称	参数类型	描述
------	------	----

Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 11-5 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

本示例用于如何设置桶的多版本状态。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
```

```
// endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketVersioning() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定桶的版本状态, 此处以 Enabled 为例
      VersionStatus: 'Enabled'
    };

    // 设置桶的多版本状态
    const result = await obsClient.setBucketVersioning(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s versioning status successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

setBucketVersioning();
```

11.3 查看桶多版本状态(Node.js SDK)

须知

开发过程中, 您有任何问题可以在github上[提交issue](#)。

功能介绍

OBS支持多版本控制, 您可以在一个桶中保留对象的多个版本, 使您更方便地检索和还原各个版本, 在意外操作或应用程序故障时快速恢复数据。更多多版本相关信息请参见[多版本控制](#)。

调用获取桶的多版本状态接口, 您可以获取指定桶的多版本状态。

接口约束

- 您必须是桶拥有者或拥有获取桶的多版本状态的权限, 才能获取桶的多版本状态。建议使用IAM或桶策略进行授权, 如果使用IAM则需授予obs:bucket:GetBucketVersioning权限, 如果使用桶策略则需授予GetBucketVersioning权限。相关授权方式介绍可参见[OBS权限控制概述](#), 配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系, 详细信息请参见[地区与终端节点](#)。

方法定义

ObsClient.getBucketVersioning(params)

请求参数

表 11-6 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个人区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

返回结果

表 11-7 返回结果

参数类型	描述
<p>表11-8 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表11-8。</p>

表 11-8 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表11-10	参数解释: 操作成功后的结果数据，详见 表11-10 。 约束限制: 当Status大于300时为空。

表 11-9 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 11-10 GetBucketVersioningOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
VersionStatus	string	参数解释: 桶的多版本状态。

代码示例

您可以通过ObsClient.getBucketVersioning查看桶的多版本状态。以下代码展示了如何查看名为examplebucket桶的多版本状态。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketVersioning() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
    };
    // 获取桶的多版本状态
    const result = await obsClient.getBucketVersioning(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s versioning status successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("VersionStatus: %s", result.InterfaceResult.VersionStatus);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

getBucketVersioning();
```

11.4 获取多版本对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

您可以根据需要将存储在OBS中对象的某一个版本下载到本地。

接口约束

- 您必须是桶拥有者或拥有下载对象的权限，才能下载对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:GetObject权限，如果使用桶策略则需授予GetObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 对于存储类别为归档存储的对象，需要确认对象的状态为“已恢复”才能对其进行下载。

方法定义

```
ObsClient.getObject(params)
```

请求参数

表 11-11 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
VersionId	string	可选	<p>参数解释: 对象的版本号。例如：G001117FCE89978B0000401205D5DC9A。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
SaveAsFile	string	可选	<p>参数解释: 下载对象的目标路径，包含文件名。</p> <p>约束限制: 包含文件名，如aa/bb.txt。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
SaveAsStream	boolean	可选	<p>参数解释: 是否将对象以流 (stream.Readable) 的形式返回。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • true: 将对象以流 (stream.Readable) 的形式返回。 • false: 不将对象以流 (stream.Readable) 的形式返回。 <p>默认取值: false</p>
ProgressCallback	function	可选	<p>参数解释: 获取下载进度的回调函数。</p> <p>说明 该回调函数依次包含三个参数: 已下载的字节数、总字节数、已使用的时间 (单位: 秒)。</p> <p>约束限制: 无</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
IfMatch	string	可选	<p>参数解释: 指定一个预设的Etag值, 如果下载对象的Etag值与该参数值相同, 则返回对象内容, 否则返回错误。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
IfModifiedSince	string	可选	<p>参数解释: 如果对象的修改时间晚于该参数值指定的时间, 则返回对象内容, 否则返回异常码。</p> <p>约束限制: 该参数值必须符合http://www.ietf.org/rfc/rfc2616.txt规定的HTTP时间格式。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
IfNoneMatch	string	可选	<p>参数解释: 指定一个预设的Etag值, 如果下载对象的ETag值与该参数值不相同, 则返回对象内容, 否则返回错误。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
IfUnmodifiedSince	string	可选	<p>参数解释: 如果对象的修改时间早于该参数值指定的时间, 则返回对象内容, 否则返回异常码。</p> <p>约束限制: 该参数值必须符合http://www.ietf.org/rfc/rfc2616.txt规定的HTTP时间格式。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Range	string	可选	<p>参数解释: 指定下载的范围。例如: 0-999, 即对象的第1个字节到1000个字节。</p> <p>约束限制: 如果Range的最大长度超出对象长度-1, 仍旧取对象长度-1。</p> <p>取值范围: 取值区间: [0, 对象长度-1], 格式: bytes=x-y。</p> <p>默认取值: 无</p>
Origin	string	可选	<p>参数解释: 预请求指定的跨域请求Origin (通常为域名)。</p> <p>约束限制: 每个匹配规则允许使用最多一个“*”通配符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
RequestHeader	string	可选	<p>参数解释: 跨域请求可以使用的HTTP头域。只有匹配上允许的头域中的配置, 才被视为是合法的CORS请求。</p> <p>约束限制: 每个头域可填写一个“*”通配符, 不支持&、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
ResponseCacheControl	string	可选	<p>参数解释: 重写响应中的Cache-Control头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Cache-Control的取值。</p> <p>默认取值: 无</p>
ResponseContentDisposition	string	可选	<p>参数解释: 重写响应中的Content-Disposition头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Disposition的取值。</p> <p>默认取值: 无</p>
ResponseContentEncoding	string	可选	<p>参数解释: 重写响应中的Content-Encoding头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Encoding的取值。</p> <p>默认取值: 无</p>
ResponseContentLanguage	string	可选	<p>参数解释: 重写响应中的Content-Language头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Content-Language的取值。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
ResponseContentType	string	可选	<p>参数解释: 重写响应中的Content-Type头。</p> <p>约束限制: 无</p> <p>取值范围: 常见的Content-Type (MIME) 列表 参见如何理解Content-Type (MIME) (Node.js SDK)</p> <p>默认取值: 无</p>
ResponseExpires	string	可选	<p>参数解释: 重写响应中的Expires头。</p> <p>约束限制: 无</p> <p>取值范围: 参见HTTP标准头域Expires的取值。</p> <p>默认取值: 无</p>
ImageProcess	string	可选	<p>参数解释: 图片处理参数, 描述针对对象的图片处理命令或处理样式。例如表示对图片依次进行缩放、旋转, 取值: image/resize,m_fixed,w_100,h_100/rotate,90。</p> <p>约束限制: 无</p> <p>取值范围: 命令方式: image/命令参数。 样式方式: style/样式名称。 详细参数说明参见处理图片。</p> <p>默认取值: 如果不输入处理命令, 将返回原图。</p>

参数名称	参数类型	是否必选	描述
SseC	string	当使用SSE-C方式时，必选	<p>参数解释： 以SSE-C方式解密对象。</p> <p>约束限制： 仅支持AES256。</p> <p>取值范围： 可选值：AES256。</p> <p>默认取值： 无</p>
SseCKey	string	当使用SSE-C方式时，必选	<p>参数解释： SSE-C方式下解密的密钥，该密钥用于解密对象。</p> <p>约束限制： 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw=</p> <p>取值范围： 无</p> <p>默认取值： 无</p>

📖 说明

- 如果包含IfUnmodifiedSince并且不符合或者包含IfMatch并且不符合，抛出异常中HTTP状态码为：412 precondition failed。
- 如果包含IfModifiedSince并且不符合或者包含IfNoneMatch并且不符合，抛出异常中HTTP状态码为：304 Not Modified。

返回结果

表 11-12 返回结果

参数类型	描述
<p>表11-13</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表11-13。</p>

表 11-13 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表11-15	参数解释: 操作成功后的结果数据，详见 表11-15 。 约束限制: 当Status大于300时为空。

表 11-14 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 11-15 GetObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
DeleteMarker	string	参数解释: 标识删除的对象是否是删除标记。
LastModified	string	参数解释: 对象的最近一次修改时间。
ContentLength	string	参数解释: 对象数据的长度。
CacheControl	string	参数解释: 响应中的Cache-Control头。
ContentDisposition	string	参数解释: 响应中的Content-Disposition头。
ContentEncoding	string	参数解释: 响应中的Content-Encoding头
ContentLanguage	string	参数解释: 响应中的Content-Language头
ContentType	string	参数解释: 对象的MIME类型。
Expires	string	参数解释: 响应中的Expires头。
ETag	string	参数解释: 对象的ETag值。
VersionId	string	参数解释: 对象的版本号。
WebsiteRedirectLocation	string	参数解释: 当桶设置了Website配置, 该参数指明对象的重定向地址。
StorageClass	string	参数解释: 对象的存储类型, 当对象存储类型是标准存储时, 该值为空。
Restore	string	参数解释: 归档存储类型对象的恢复状态。
Expiration	string	参数解释: 对象的详细过期信息。

参数名称	参数类型	描述
Content	string stream.Readable	参数解释: 对象的内容。当设置了SaveAsFile时该值为空；当设置了SaveAsStream且为true时该值为stream.Readable对象；当SaveAsFile和SaveAsStream都未设置时为Buffer对象。
Metadata	object	参数解释: 对象自定义元数据。需要在桶的CORS配置中增加允许响应中可返回的附加头域。例如，新增x-amz-meta-property1以获取自定义元数据property1。

代码示例

您可以通过ObsClient.getObject接口指定VersionId参数来获取多版本对象，代码示例如下：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: 'example/objectname',
      // 指定版本对象的版本号，如果版本号为空则默认下载最新版本的对象
      VersionId: 'G001117FCE89978B0000401205D5DC9A'
    };
  }

  // 下载版本对象
  const result = await obsClient.getObject(params);
  if (result.CommonMsg.Status <= 300) {
    console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    return;
  }
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
  console.log("Status: %d", result.CommonMsg.Status);
}
```

```
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};
getObject();
```

11.5 复制多版本对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

用户可以根据需要将存储在OBS上的多版本对象复制到其他路径下，为指定桶中的多版本对象创建一个副本。在单次操作中，您可以创建最大5GB的对象副本。

接口约束

- 您必须是桶拥有者或拥有复制对象的权限，才能复制对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:PutObject权限，如果使用桶策略则需授予PutObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 复制对象操作的请求需要通过头域携带复制的源桶和对象信息，不能携带消息实体。
- 目标对象大小范围是[0, 5GB]，如果源对象大小超过5GB，只能使用[复制段\(Node.js SDK\)](#)功能复制部分对象。

方法定义

```
ObsClient.copyObject(params)
```

请求参数

表 11-16 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p>
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
CopySource	string	必选	<p>参数解释: 用来指定复制对象操作的源桶名以及源对象名。当源对象存在多个版本时,通过versionId参数指定版本源对象。</p> <p>示例: source_bucket/sourceObject?versionId=G001117FCE89978B0000401205D5DC9A</p> <p>约束限制: 无</p> <p>约束限制: 中文字符与%, 需要进行URL编码。</p> <p>默认取值: 无</p>
ACL	AclType	可选	<p>参数解释: 复制对象时可指定的预定义访问策略。有关访问控制列表 (Access Control List, ACL) 功能的详细信息可参见ACL功能介绍。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的访问策略选项参见AclType。</p> <p>默认取值: 无</p>
MetadataDirective	MetadataDirectiveType	可选	<p>参数解释: 复制源对象属性的策略。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的对象属性选项参见MetadataDirectiveType。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
CopySourceIfMatch	string	可选	<p>参数解释: 如果源对象的ETag值与该参数值相同, 则进行复制, 否则返回错误。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
CopySourceIfNoneMatch	string	可选	<p>参数解释: 如果源对象的ETag值与该参数值不相同, 则进行复制, 否则返回错误。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无</p>
CopySourceIfUnmodifiedSince	string	可选	<p>参数解释: 如果源对象在指定的时间后没有修改, 则进行复制, 否则抛出异常。</p> <p>约束限制: 该参数值必须符合http://www.ietf.org/rfc/rfc2616.txt规定的HTTP时间格式。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
CopySourceIfModifiedSince	string	可选	<p>参数解释: 如果源对象在指定的时间后有修改, 则进行复制, 否则抛出异常。</p> <p>约束限制: 该参数值必须符合http://www.ietf.org/rfc/rfc2616.txt规定的HTTP时间格式。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
WebsiteRedirectLocation	string	可选	<p>参数解释: 当桶设置了Website配置, 可以将获取这个对象的请求重定向到桶内另一个对象或一个外部的URL。 例如, 重定向请求到桶内另一对象: WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL: WebsiteRedirectLocation:http://www.example.com/ OBS将这个值从头域中取出, 保存在对象的元数据“WebsiteRedirectLocation”中。</p> <p>约束限制:</p> <ul style="list-style-type: none">• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。• OBS仅支持为桶根目录下的对象设置重定向, 不支持为桶中文件夹下的对象设置重定向。 <p>取值范围: 无</p> <p>默认取值: 无</p>
CopySourceSseC	string	当使用SSE-C方式时, 必选	<p>参数解释: 表示服务端加密是SSE-C方式。对象使用SSE-C加密方式。</p> <p>约束限制: 仅支持AES256。</p> <p>取值范围: 可选值: AES256, 即选择SSE-C方式加密对象。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
CopySourceSseCKey	string	当使用SSE-C方式时，必选	<p>参数解释： SSE-C方式下加密使用的密钥，该密钥用于加密对象。</p> <p>约束限制： 该头域由256bit的密钥经过Base64编码得到，示例： K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw=</p> <p>取值范围： 无</p> <p>默认取值： 无</p>
CacheControl	string	可选	<p>参数解释： 复制时重写响应中的Cache-Control头。</p> <p>约束限制： 无</p> <p>取值范围： 参见HTTP标准头域Cache-Control的取值。</p> <p>默认取值： 无</p>
ContentDisposition	string	可选	<p>参数解释： 复制时重写响应中的Content-Disposition头。</p> <p>约束限制： 无</p> <p>取值范围： 参见HTTP标准头域Content-Disposition的取值。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
ContentEncoding	string	可选	参数解释: 复制时重写响应中的Content-Encoding头。 约束限制: 无 取值范围: 参见HTTP标准头域Content-Encoding的取值。 默认取值: 无
ContentLanguage	string	可选	参数解释: 复制时重写响应中的Content-Language头。 约束限制: 无 取值范围: 参见HTTP标准头域Content-Language的取值。 默认取值: 无
ContentType	string	可选	参数解释: 复制时重写响应中的Content-Type头。 约束限制: 无 取值范围: 常见的Content-Type (MIME) 列表参见 如何理解Content-Type (MIME) (Node.js SDK) 默认取值: 无
Expires	string	可选	参数解释: 复制时重写响应中的Expires头。 约束限制: 无 取值范围: 参见HTTP标准头域Expires的取值。 默认取值: 无

参数名称	参数类型	是否必选	描述
StorageClass	StorageClassType	可选	<p>参数解释: 复制时设置对象的存储类别。</p> <p>约束限制: 无</p> <p>取值范围: 可选择的存储类型参见 StorageClassType。</p> <p>默认取值: 无, 如果未设置此头域, 则对象的存储类型默认为桶的存储类型。</p>
Metadata	object	可选	<p>参数解释: 目标对象的自定义元数据。OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据, 以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时, 加入的自定义元数据将会在返回的消息头中出现。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 所有自定义元数据大小的总和不超过8K。单个自定义元数据大小的计算方式为: 每个键和值的UTF-8 编码中的字节总数。 • 自定义元数据的key值不区分大小写, OBS统一转为小写进行存储。value值区分大小写。 • 自定义元数据key-value对都必须符合US-ASCII。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。 <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantRead	string	可选	<p>参数解释: 被授予READ权限的账号的账号ID, 即 domain_id。账号被授予READ权限后, 针对当前正在复制的对象, 允许获取对象的内容、获取对象的元数据。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantReadAcp	string	可选	<p>参数解释: 被授予READ_ACP权限的账号的账号ID, 即domain_id。账号被授予READ_ACP权限后, 针对当前正在复制的对象, 允许读对象的ACL信息。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
GrantWriteAcp	string	可选	<p>参数解释: 被授予WRITE_ACP权限的账号的账号ID, 即domain_id。账号被授予WRITE_ACP权限后, 针对当前正在复制的对象, 允许修改对象的ACL信息。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
GrantFullControl	string	可选	<p>参数解释: 被授予FULL_CONTROL权限的账号的账号ID, 即domain_id。账号被授予FULL_CONTROL权限后, 针对当前正在复制的对象, 允许进行任何操作, 读对象、获取对象元数据、获取对象ACL、写对象ACL。</p> <p>约束限制: 如果授权给多个租户, 需要通过“,”分隔。</p> <p>取值范围: 获取账号ID的方法请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
SuccessActionRedirect	string	可选	<p>参数解释: 此参数的值是一个URL, 用于指定当此次请求操作成功响应后的重定向的地址。如果此参数值有效且操作成功, 响应码为303, Location头域由此参数以及桶名、对象名、对象的ETag组成。如果此参数值无效, 则OBS忽略此参数的作用, Location头域为对象地址, 响应码根据操作成功或失败正常返回。</p> <p>约束限制: 无</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

表 11-17 AclType

常量名	原始值	说明
ObsClient.enums.AclPrivate	private	私有读写。 桶或对象的所有者拥有完全控制的权限, 其他任何人都没有访问权限。

常量名	原始值	说明
ObsClient.enums.AclPublicRead	public-read	公共读私有写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadWrite	public-read-write	公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务。 设在对象上，所有人可以获取该对象内容和元数据。
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	桶公共读，桶内对象公共读。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadDelivered不能应用于对象。
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	桶公共读写，桶内对象公共读写。 设在桶上，所有人可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本、上传对象、删除对象、初始化段任务、上传段、合并段、复制段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 说明 AclPublicReadWriteDelivered不能应用于对象。

常量名	原始值	说明
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	<p>设在对象上，桶和对象的所有者拥有对象的完全控制权限，其他任何人都没有访问权限。</p> <p>默认情况下，上传对象至其他用户的桶中，桶拥有者没有对象的控制权限。对象拥有者为桶拥有者添加此权限控制策略后，桶所有者可以完全控制对象。例如，用户A上传对象x至用户B的桶中，系统默认用户B没有对象x的控制权。当用户A为对象x设置bucket-owner-full-control策略后，用户B就拥有了对象x的控制权。</p>

表 11-18 MetadataDirectiveType

常量名	原始值	说明
ObsClient.enums.CopyMetadata	COPY	<p>复制对象时，复制源对象属性。</p> <p>须知 该值仅在复制对象(Node.js SDK)接口中使用。</p>
ObsClient.enums.ReplaceMetadata	REPLACE	<p>使用当前请求中携带的头域完整替换，未指定的元数据会被删除。</p>
ObsClient.enums.ReplaceNewMetadata	REPLACE_NEW	<p>对于已经存在值的元数据进行替换，不存在值的元数据进行赋值，未指定的元数据保持不变。</p> <p>须知 该值仅在设置对象元数据(Node.js SDK)接口中使用。</p>

表 11-19 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。</p>

常量名	原始值	说明
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

📖 说明

- 如果包含CopySourceIfUnmodifiedSince并且不符合，或者包含CopySourceIfMatch并且不符合，或者包含CopySourceIfModifiedSince并且不符合，或者包含CopySourceIfNoneMatch并且不符合，抛出异常中HTTP状态码为：412 precondition failed。
- CopySourceIfModifiedSince和CopySourceIfNoneMatch可以一起使用；CopySourceIfUnmodifiedSince和CopySourceIfMatch可以一起使用。

返回结果

表 11-20 返回结果

参数类型	描述
表11-21 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见 表11-21 。

表 11-21 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表11-23	参数解释： 操作成功后的结果数据，详见 表11-23 。 约束限制： 当Status大于300时为空。

表 11-22 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 11-23 CopyObjectOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
LastModified	string	参数解释: 目标对象的最近一次修改时间（UTC时间）。

参数名称	参数类型	描述
ETag	string	<p>参数解释: 目标对象的ETag值。对象的base64编码的128位MD5摘要。ETag是对象内容的唯一标识，可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A，下载对象时ETag为B，则说明对象内容发生了变化。ETag只反映变化的内容，而不是其元数据。上传的对象或复制操作创建的对象，都有唯一的ETag。</p> <p>约束限制: 当对象是服务端加密的对象时，ETag值不是对象的MD5值。</p>
CopySourceVersionId	string	<p>参数解释: 源对象的版本号。</p>
VersionId	string	<p>参数解释: 目标对象的版本号。</p>

代码示例

您可以通过ObsClient.copyObject接口在CopySource参数中指定待复制对象的versionId来复制多版本对象，代码示例如下：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyObject() {
  try {
    const params = {
      // 指定目标存储桶名称
      Bucket: "examplebucket",
      // 指定目标对象名，此处以 example/objectname 为例
      Key: 'example/objectname',
      // 指定复制源桶名、源对象名和源对象版本ID，此处复制源桶名以sourcebucketname为例，复制源对象名以sourceobjectname为例，版本ID以G001117FCE89978B0000401205D5DC9A为例
      CopySource: 'sourcebucketname/sourceobjectname?versionId=G001117FCE89978B0000401205D5DC9A'
    };
    // 复制版本对象
    const result = await obsClient.copyObject(params);
  }
}
```

```
if (result.CommonMsg.Status <= 300) {
    console.log("Copy Object(bucket:%s, object: %s) successful from bucket/object: %s!", params.Bucket,
params.Key, params.CopySource);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    console.log("ETag: %s, LastModified:%s", result.InterfaceResult.ETag,
result.InterfaceResult.LastModified);
    return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
};
};
copyObject();
```

11.6 恢复多版本归档存储对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能说明

如果要获取归档存储对象的内容，需要先将对象恢复，然后再执行下载数据的操作。对象恢复后，会产生一个标准存储类型的对象副本，也就是说会同时存在标准存储类型的对象副本和归档存储对象，在恢复对象的保存时间到期后标准存储类型的对象副本会自动删除。

该接口可以恢复指定桶中的归档存储对象。

接口约束

- 您必须是桶所有者或拥有恢复归档存储对象的权限，才能恢复归档存储对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object:RestoreObject权限，如果使用桶策略则需授予RestoreObject权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[配置对象策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 重复恢复归档存储对象时在延长恢复有效期的同时，也将会对恢复时产生的恢复费用进行重复收取。产生的标准存储类别的对象副本有效期将会延长，并且收取延长时间段产生的标准存储副本费用。

方法定义

```
ObsClient.restoreObject(params)
```

请求参数

表 11-24 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
Key	string	必选	<p>参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。 例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p>约束限制: ObsClient.restoreObject中指定的对象必须是归档存储类别，否则调用该接口会报错。</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
Version Id	string	可选	<p>参数解释: 待恢复归档存储对象的版本号。</p> <p>约束限制: 无</p> <p>取值范围: 长度为32的字符串。</p> <p>默认取值: 无，如果不设置则默认指定最新版本的对象。</p>
Days	number	必选	<p>参数解释: 恢复对象后，会生成一个对象的标准存储副本，此参数指定恢复有效期，即标准存储副本的保存时间。</p> <p>约束限制: 无</p> <p>取值范围: [1, 30]，单位：天。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Tier	RestoreTierType	可选	<p>参数解释: 恢复选项，用户可根据需求选择恢复选项，不同的恢复选项恢复耗时不同。</p> <p>约束限制: 无</p> <p>取值范围: 恢复选项可选值详见RestoreTierType。</p> <p>默认取值: Standard，即标准恢复。</p>

表 11-25 RestoreTierType

常量名	原始值	说明
ObsClient.enums.RestoreTierExpedited	Expedited	快速恢复，归档存储恢复耗时1~5 min。
ObsClient.enums.RestoreTierStandard	Standard	标准恢复，归档存储恢复耗时3~5 h。

返回结果

表 11-26 返回结果

参数类型	描述
<p>表11-27</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表 11-27。</p>

表 11-27 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	<p>参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICommonMsg。</p>

参数名称	参数类型	描述
InterfaceResult	表11-29	参数解释: 操作成功后的结果数据, 详见表11-29。 约束限制: 当Status大于300时为空。

表 11-28 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 11-29 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

本示例用于在恢复examplebucket桶中对象名称为example/objectname的归档存储对象，该对象版本号为G001117FCE89978B0000401205D5DC9A。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function restoreObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定版本对象的版本号
      VersionId: 'G001117FCE89978B0000401205D5DC9A'
      // 指定恢复对象的保存时间，此处以1为例，单位天，取值范围：[1, 30]
      Days: 1,
      // 指定恢复选项，此处以obs.RestoreTierExpedited为例，默认为快速恢复
      Tier: obs.enums.RestoreTierExpedited
    };
    // 取回归档存储对象
    const result = await obsClient.restoreObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Restore object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

restoreObject();
```

注意

重复恢复归档存储数据时在延长恢复有效期的同时，也将会对恢复时产生的恢复费用进行重复收取。产生的标准存储类别的对象副本有效期将会延长，并且收取延长时间段产生的标准存储副本费用。

11.7 列举桶内多版本对象(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

用列举桶内对象接口，可列举指定桶内的部分或所有多版本对象的描述信息。您可以通过设置前缀、数量、起始位置等参数，返回符合您筛选条件的多版本对象信息。返回结果以多版本对象名的字典序排序。

接口约束

- 您必须是桶拥有者或拥有列举桶内多版本对象的权限，才能列举桶内多版本对象。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:ListBucketVersions权限，如果使用桶策略则需授予ListBucketVersions权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.listVersions(params)
```

请求参数

表 11-30 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my.bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>
KeyMarker	string	可选	<p>参数解释: 列举多版本对象的起始位置，返回的对象列表将是对象名按照字典序排序后该参数以后的所有对象。</p> <p>约束限制: 该字段仅用于多版本列举。</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
VersionIdMarker	string	可0选	<p>参数解释: 与KeyMarker配合使用。返回的对象列表将是对象名和版本号按照字典序排序后该参数以后的所有对象。KeyMarker指定对象名, VersionIdMarker指定该对象的具体版本号, 两者共同确定对象版本</p> <p>约束限制:</p> <ul style="list-style-type: none"> 该字段只适用于多版本列举场景。 如果VersionIdMarker指定的版本号不属于KeyMarker指定对象的版本, 则该参数无效。 <p>取值范围: 对象的版本号, 即KeyMarker对象的某一个VersionId值。</p> <p>默认取值: 无</p>
Prefix	string	可选	<p>参数解释: 限定返回的对象名必须带有指定前缀。</p> <p>例如, 假设您拥有以下对象: logs/day1、logs/day2、logs/day3和ExampleObject.jpg。如果您将logs/指定为前缀, 将返回以字符串“logs/”开头的三个对象。如果您指定空的前缀且请求中没有其他过滤条件, 将返回桶中的所有对象。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
MaxKeys	number	可选	<p>参数解释: 列举对象的最大数目, 返回的对象列表将是按照字典顺序的最多前MaxKeys个对象。</p> <p>约束限制: 无</p> <p>取值范围: 1~1000, 当超出范围时, 按照默认的1000进行处理。</p> <p>默认取值: 1000</p>

参数名称	参数类型	是否必选	描述
Delimiter	string	可选	<p>参数解释:</p> <p>对象名按照此标识符进行分组。通常与Prefix参数搭配使用，如果指定了Prefix，从Prefix到第一次出现Delimiter间具有相同字符串的对象名会被分成一组，形成一条CommonPrefixes；如果没有指定Prefix，从对象名的首字符到第一次出现Delimiter间具有相同字符串的对象名会被分成一组，形成一条CommonPrefixes。</p> <p>例如，桶中有3个对象，分别为abcd、abcde、bbcde。如果指定Delimiter为d，Prefix为a，abcd、abcde会被分成一组，形成一条前缀为abcd的CommonPrefixes；如果只指定Delimiter为d，abcd、abcde会被分成一组，形成一条前缀为abcd的CommonPrefixes，而bbcde会被单独分成一组，形成一条前缀为bbcd的CommonPrefixes。</p> <p>对于并行文件系统，不携带此参数时默认列举是递归列举此目录下所有内容，会列举子目录。在大数据场景下（目录层级深、目录下文件多）的列举，建议设置[Delimiter=/]，只列举当前目录下的内容，不列举子目录，提高列举效率。</p> <p>约束限制:</p> <p>无</p> <p>取值范围:</p> <p>长度大于0且不超过1024的字符串。</p> <p>默认取值:</p> <p>无</p>
EncodingType	string	可选	<p>参数解释:</p> <p>对响应中的部分元素进行指定类型的编码。如果Delimiter、KeyMarker、Prefix、CommonPrefixes、NextKeyMarker和Key包含xml 1.0标准不支持的控制字符，可通过设置该参数对响应中的Delimiter、KeyMarker、Prefix、CommonPrefixes、NextKeyMarker和Key进行编码。</p> <p>约束限制:</p> <p>无</p> <p>取值范围:</p> <p>可选值为url。</p> <p>默认取值:</p> <p>无，不设置则不编码。</p>

返回结果

表 11-31 返回结果

参数类型	描述
表11-32 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见表11-32。

表 11-32 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	ListVersionsOutput	参数解释： 操作成功后的结果数据，详见 ListVersionsOutput 。 约束限制： 当Status大于300时为空。

表 11-33 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释： OBS服务端返回的HTTP状态码。 取值范围： 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 。
Code	string	参数解释： OBS服务端返回的错误码。
Message	string	参数解释： OBS服务端返回的错误描述。
HostId	string	参数解释： OBS服务端返回的请求服务端ID。
RequestId	string	参数解释： OBS服务端返回的请求ID。

Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 11-34 ListVersionsOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。 默认取值: 无
Location	string	参数解释: 桶所在的区域。
Name	string	参数解释: 桶名。
Prefix	string	参数解释: 对象名的前缀，与请求中的该参数对应。 例如，假设您拥有以下对象：logs/day1、logs/day2、logs/day3和ExampleObject.jpg。如果您将logs/指定为前缀，将返回以字符串“logs/”开头的三个对象。如果您指定空的前缀且请求中没有其他过滤条件，将返回桶中的所有对象。
KeyMarker	string	参数解释: 列举多版本对象的起始位置，与请求中的该参数对应。返回的对象列表将是对象名按照字典序排序后该参数以后的所有对象。
VersionIdMarker	string	参数解释: keyMarker对象的对应版本号。 与KeyMarker配合使用。返回的对象列表将是对象名和版本号按照字典序排序后该参数以后的所有对象。KeyMarker指定对象名，VersionIdMarker指定该对象的具体版本号，两者共同确定对象版本

参数名称	参数类型	描述
Delimiter	string	<p>参数解释:</p> <p>对象名按照此标识符进行分组。通常与Prefix参数搭配使用, 如果指定了Prefix, 从Prefix到第一次出现Delimiter间具有相同字符串的对象名会被分成一组, 形成一条CommonPrefixes; 如果没有指定Prefix, 从对象名的首字符到第一次出现Delimiter间具有相同字符串的对象名会被分成一组, 形成一条CommonPrefixes。</p> <p>例如, 桶中有3个对象, 分别为abcd、abcde、bbcde。如果指定Delimiter为d, Prefix为a, abcd、abcde会被分成一组, 形成一条前缀为abcd的CommonPrefixes; 如果只指定Delimiter为d, abcd、abcde会被分成一组, 形成一条前缀为abcd的CommonPrefixes, 而bbcde会被单独分成一组, 形成一条前缀为bbcd的CommonPrefixes。</p> <p>对于并行文件系统, 不携带此参数时默认列举是递归列举此目录下所有内容, 会列举子目录。在大数据场景下(目录层级深、目录下文件多)的列举, 建议设置[Delimiter=/], 只列举当前目录下的内容, 不列举子目录, 提高列举效率。</p>
MaxKeys	number	<p>参数解释:</p> <p>列举对象的最大数目, 与请求中的该参数对应。返回的对象列表将是按照字典顺序的最多前MaxKeys个对象。</p>
IsTruncated	boolean	<p>参数解释:</p> <p>表明本次请求是否返回了全部结果。因为每次列举返回对象的数量上限是1000个, 如果对象个数大于1000, 则无法通过一次请求返回全部结果。</p> <p>取值范围:</p> <ul style="list-style-type: none">• true: 表示没有返回全部结果。• false: 表示已返回了全部结果。
NextKeyMarker	string	<p>参数解释:</p> <p>下次列举多版本对象请求的起始位置。如果本次没有返回全部结果, 响应请求中将包含该元素, 用于标明接下来请求的KeyMarker值。</p>
NextVersionIdMarker	string	<p>参数解释:</p> <p>下次列举多版本对象请求的起始位置 (VersionIdMarker), 与NextKeyMarker配合使用。如果本次没有返回全部结果, 响应请求中将包含该元素, 用于标明接下来请求的VersionIdMarker值。</p>
Versions	Version []	<p>参数解释:</p> <p>桶内多版本对象列表, 详见Version。</p>

参数名称	参数类型	描述
DeleteMarkers	DeleteMarker[]	参数解释: 桶内多版本删除标记列表, 详见DeleteMarker。
CommonPrefixes	表11-35[]	参数解释: 当请求中设置了Delimiter分组字符时, 返回按Delimiter分组后的对象名称前缀列表, 详见CommonPrefix。
EncodingType	string	参数解释: 表明响应中的部分元素的编码类型。如果Delimiter、KeyMarker、Prefix、CommonPrefixes、NextKeyMarker和Key包含xml 1.0标准不支持的控制字符, 可通过设置该参数对响应中的Delimiter、KeyMarker、Prefix、CommonPrefixes、NextKeyMarker和Key进行编码。

表 11-35 CommonPrefix

参数名称	参数类型	描述
Prefix	string	参数解释: 为桶内对象的前缀。

表 11-36 Version

参数名称	参数类型	描述
Key	string	参数解释: 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径, 路径中不包含桶名。 例如, 您对象的访问地址为examplebucket.obs.ap-southeast-1..com/folder/test.txt 中, 对象名为folder/test.txt。 取值范围: 长度大于0且不超过1024的字符串。 默认取值: 无

参数名称	参数类型	描述
VersionId	string	参数解释: 对象的版本号。 取值范围: 长度为32的字符串。 默认取值: 无
LastModified	string	参数解释: 对象最近一次被修改的时间（UTC时间）。 取值范围: UTC时间 默认取值: 无
ETag	string	参数解释: 对象的base64编码的128位MD5摘要。ETag是对象内容的唯一标识，可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A，下载对象时ETag为B，则说明对象内容发生了变化。ETag只反映变化的内容，而不是其元数据。上传的对象或复制操作创建的对象，都有唯一的ETag。 约束限制: 当对象是服务端加密的对象时，ETag值不是对象的MD5值。 取值范围: 长度为32的字符串。 默认取值: 无
Size	number	参数解释: 对象的字节数。 取值范围: 0~48.8TB，单位：字节。 默认取值: 无
Owner	Owner	参数解释: 对象的所有者，详见 Owner 。
StorageClass	StorageClassType	参数解释: 对象的存储类型，详见 StorageClassType 。

参数名称	参数类型	描述
IsLatest	boolean	<p>参数解释： 标识对象是否是最新的版本。</p> <p>取值范围：</p> <ul style="list-style-type: none"> • true：代表是最新的版本。 • false：代表非最新版本。 <p>默认取值： false</p>

表 11-37 DeleteMarker

参数名称	参数类型	描述
Key	string	<p>参数解释： 对象名。对象名是对象在存储桶中的唯一标识。对象名是对象在桶中的完整路径，路径中不包含桶名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p>
VersionId	string	<p>参数解释： 对象的版本号。</p>
IsLatest	boolean	<p>参数解释： 标识对象是否是最新的版本。</p> <p>取值范围：</p> <ul style="list-style-type: none"> • true：代表是最新的版本。 • false：代表非最新版本。
LastModified	string	<p>参数解释： 对象最近一次被修改的时间（UTC时间）。</p>
Owner	Owner	<p>参数解释： 对象的所有者，包含对象拥有者DomainId和对象拥有者名称，详见Owner。</p>
StorageClass	StorageClassType	<p>参数解释： 对象的存储类型，详见存储类型取值范围详见StorageClassType。</p>

表 11-38 Owner

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时必选	<p>参数解释: 所有者的账号ID, 即domain_id。</p> <p>取值范围: 如何获取账号ID请参见如何获取账号ID和用户ID? (Node.js SDK)</p> <p>默认取值: 无</p>
Display Name	string	可选	<p>参数解释: 所有者的账号用户名。</p> <p>默认取值: 无</p>

表 11-39 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量, 适用于有大量热点对象(平均一个月多次)或小对象(<1MB), 且需要频繁访问数据的业务场景。</p>
ObsClient.enums.StorageClassWarm	WARM	<p>低频访问存储。</p> <p>低频访问存储适用于不频繁访问(平均一年少于12次)但在需要时也要求能够快速访问数据的业务场景。</p>
ObsClient.enums.StorageClassCold	COLD	<p>归档存储。</p> <p>归档存储适用于很少访问(平均一年访问一次)数据的业务场景。</p>

代码示例一：简单列举

本示例用于简单列举多版本对象, 最多返回1000个对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-
```

```
cn/usermanual-ca/ca_01_0003.html
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露
// 风险。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
// security_token: process.env.SECURITY_TOKEN,
// endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listVersions() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
    };
    // 列举桶内版本对象
    const result = await obsClient.listVersions(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      // 获取多版本对象
      for (let j = 0; j < result.InterfaceResult.Versions.length; j++) {
        const val = result.InterfaceResult.Versions[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',
          j, val.Owner.ID, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);
      };
      // 获取对象删除标记
      for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {
        const val = result.InterfaceResult.Versions[j];
        console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
          j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
      };
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

listVersions();
```

📖 说明

- 每次至多返回1000个多版本对象，如果指定桶包含的对象数量大于1000，则返回结果中InterfaceResult.IsTruncated为true表明本次没有返回全部对象，并可通过InterfaceResult.NextKeyMarker和InterfaceResult.NextVersionIdMarker获取下次列举的起始位置。
- 如果想获取指定桶包含的所有多版本对象，可以采用[代码示例五：分页列举全部多版本对象](#)的方式。

代码示例二：指定数目列举

本示例用于指定数目列举多版本对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
```

```
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listVersions() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定返回的最大对象数，此处以 100 为例，返回的对象列表将是按照字典顺序的最多前max-keys个对象
      MaxKeys: 100,
    };
    // 列举桶内版本对象
    const result = await obsClient.listVersions(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List version objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        // 获取多版本对象
        for (let i = 0; i < result.InterfaceResult.Versions.length; i++) {
          const val = result.InterfaceResult.Versions[i];
          console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',
            j, val.Owner.ID, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);
        };
        // 获取对象删除标记
        for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {
          const val = result.InterfaceResult.Versions[i];
          console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
            j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
        };
      };
      return;
    };
  } catch (error) {
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

listVersions();
```

代码示例三：指定前缀列举

本示例用于指定前缀列举多版本对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');
```



```
// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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
  // 您可以通过登录访问管理控制台获取访问密钥AK/SK, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listVersions() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定列举对象前缀, 此处以“test/”前缀为例, 满足指定前缀的对象会被列举
      Prefix: "test/",
      // 指定返回的最大对象数, 此处以 100 为例, 返回的对象列表将是按照字典顺序的最多前max-keys个对象, 默认值为1000
      MaxKeys: 100,
    };

    // 列举桶内版本对象
    const result = await obsClient.listVersions(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List version objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        // 获取多版本对象
        for (let j = 0; j < result.InterfaceResult.Versions.length; j++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',
            j, val.Owner.ID, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);
        };
        // 获取对象删除标记
        for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
            j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
        };
      };
      return;
    };
  };

  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
  console.log("Status: %d", result.CommonMsg.Status);
  console.log("Code: %s", result.CommonMsg.Code);
  console.log("Message: %s", result.CommonMsg.Message);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};

listVersions();
```

代码示例四：指定起始位置列举

本示例用于指定起始位置列举多版本对象。

```
// 引入obs库
// 使用npm安装
```

```
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listVersions() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定列举对象前缀，此处以“test/”前缀为例，满足指定前缀的对象会被列举
      Prefix: "test/",
      // 指定返回的最大对象数，此处以100为例，返回的对象列表将是按照字典顺序的最多前max-keys个对象
      MaxKeys: 100,
      // 指定列举对象的起始位置，此处以“test/test2”为例
      KeyMarker: "test/test2",
    };
    // 列举桶内版本对象
    const result = await obsClient.listVersions(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List version objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        // 获取多版本对象
        for (let j = 0; j < result.InterfaceResult.Versions.length; j++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',
            j, val.Owner.ID, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);
        };
        // 获取对象删除标记
        for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
            j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
        };
      };
      return;
    };
  };
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
  console.log("Status: %d", result.CommonMsg.Status);
  console.log("Code: %s", result.CommonMsg.Code);
  console.log("Message: %s", result.CommonMsg.Message);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
listVersions();
```

代码示例五：分页列举全部多版本对象

本示例用于分页列举全部多版本对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listVersions() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定编码方式，此处以“url”为例，如果列举对象中存在特殊字符，则该参数必传
      EncodingType: "url",
    };
    while (true) {
      // 列举桶内版本对象
      const result = await obsClient.listVersions(params);
      if (result.CommonMsg.Status > 300) {
        console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
        console.log("Status: %d", result.CommonMsg.Status);
        console.log("Code: %s", result.CommonMsg.Code);
        console.log("Message: %s", result.CommonMsg.Message);
        console.log("RequestId: %s", result.CommonMsg.RequestId);
        return;
      }
      console.log("List version objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        // 获取多版本对象
        for (let j = 0; j < result.InterfaceResult.Versions.length; j++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',
            j, val.Owner.ID, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);
        }
        // 获取对象删除标记
        for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
            j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
        }
      }
      if (result.InterfaceResult.IsTruncated === "true") {
        params.KeyMarker = result.InterfaceResult.NextKeyMarker;
        params.VersionIdMarker = result.InterfaceResult.NextVersionIdMarker;
      } else {
        break;
      }
    }
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}
```

```
listVersions();
```

代码示例六：列举文件夹中的所有多版本对象

OBS本身是没有文件夹的概念的，桶中存储的元素只有对象。文件夹对象实际上是一个大小为0且对象名以“/”结尾的对象，将这个文件夹名作为前缀，即可模拟列举文件夹中对象的功能。以下代码展示如何列举文件夹中的多版本对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listVersions() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定列举对象前缀，此处以“test/”前缀为例，满足指定前缀的对象会被列举
      Prefix: "test/",
      // 指定编码方式，此处以“url”为例，如果列举对象中存在特殊字符，则该参数必传
      EncodingType: "url",
    };
    while (true) {
      // 列举桶内版本对象
      const result = await obsClient.listVersions(params);
      if (result.CommonMsg.Status > 300) {
        console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
        console.log("Status: %d", result.CommonMsg.Status);
        console.log("Code: %s", result.CommonMsg.Code);
        console.log("Message: %s", result.CommonMsg.Message);
        console.log("RequestId: %s", result.CommonMsg.RequestId);
        return;
      };
      console.log("List version objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        // 获取多版本对象
        for (let j = 0; j < result.InterfaceResult.Versions.length; j++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',
            j, val.Owner.ID, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);
        };
        // 获取对象删除标记
        for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
            j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
        };
      };
      if (result.InterfaceResult.IsTruncated === "true") {

```

```
params.KeyMarker = result.InterfaceResult.NextKeyMarker;
params.VersionIdMarker = result.InterfaceResult.NextVersionIdMarker;
} else {
    break;
};
};
} catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
};
};
listVersions();
```

11.8 设置多版本对象权限(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

可以通过ObsClient.setObjectAcl接口指定VersionId参数设置多版本对象的访问权限，接口定义请参见[设置对象ACL\(Node.js SDK\)](#)。

代码示例

本示例用于设置examplebucket桶下对象example/objectname的ACL权限控制为所有用户拥有公共读的权限，指定用户（0a03f5833900d3730f13c00f49d5exxx）拥有写的权限。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
    // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
    access_key_id: process.env.ACCESS_KEY_ID,
    secret_access_key: process.env.SECRET_ACCESS_KEY,
    // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
    // security_token: process.env.SECURITY_TOKEN,
    // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
    server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setObjectAcl() {
    try {
        const params = {
            // 指定存储桶名称
            Bucket: "examplebucket",
            // 指定对象名，此处以 example/objectname 为例
            Key: "example/objectname",
            // 指定版本对象的版本号
            VersionId: 'G001117FCE89978B0000401205D5DC9A',
```

```
// 指定对象的所有者信息
Owner: { 'ID': 'ownerid' },
// 指定被授权的用户信息
Grants: [
  // 为指定用户设置写权限, 此处用户ID为0a03f5833900d3730f13c00f49d5exxx
  { Grantee: { Type: 'CanonicalUser', ID: '0a03f5833900d3730f13c00f49d5exxx' }, Permission:
obsClient.enums.PermissionWrite },
  // 为所有用户设置读权限
  { Grantee: { Type: 'Group', URL: obsClient.enums.GroupAllUsers }, Permission:
obsClient.enums.PermissionRead },
]
};
// 设置版本对象ACL
const result = await obsClient.setObjectAcl(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Set Object(%s)'s acl successful with Bucket(%s)!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  return;
}
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
}
};
};
setObjectAcl();
```

📖 说明

- 设置多版本对象权限过程中返回的错误码含义、问题原因及处理措施可参考[OBS服务端错误码\(Node.js SDK\)](#)

11.9 获取多版本对象权限(Node.js SDK)

须知

开发过程中, 您有任何问题可以在github上[提交issue](#)。

功能介绍

可以通过ObsClient.getObjectAcl接口指定VersionId参数获取多版本对象的访问权限, 接口定义请参见[获取对象ACL\(Node.js SDK\)](#)。

代码示例

本示例用于获取名为examplebucket桶下example/objectname 版本对象的ACL权限控制列表信息, 对应的版本号是 G001117FCE89978B0000401205D5DC9A。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');
```

```
// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObjectAcl() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名, 此处以 example/objectname 为例
      Key: 'example/objectname',
      // 指定版本对象的版本号
      VersionId: 'G001117FCE89978B0000401205D5DC9A'
    };
    // 获取版本对象ACL
    const result = await obsClient.getObjectAcl(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s)'s acl successful with bucket(%s)!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("Owner[ID]: %s", result.InterfaceResult.Owner.ID);
      console.log("Owner[Name]: %s", result.InterfaceResult.Owner.Name);
      for (let i = 0; i < result.InterfaceResult.Grants.length; i++) {
        const grant = result.InterfaceResult.Grants[i];
        fmt.Printf("Grant[%d]-Type:%s, ID:%s, URI:%s, Permission:%s\n",
          index, grant.Grantee.Type, grant.Grantee.ID, grant.Grantee.URI, grant.Permission);
      }
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

getObjectAcl();
```

说明

- 设置多版本对象权限过程中返回的错误码含义、问题原因及处理措施可参考[OBS服务端错误码\(Node.js SDK\)](#)

11.10 删除多版本对象(Node.js SDK)

须知

开发过程中, 您有任何问题可以在github上[提交issue](#)。

功能介绍

可以通过ObsClient.deleteObject接口指定VersionId参数删除多版本对象，接口定义请参见[删除对象\(Node.js SDK\)](#)。

可以通过ObsClient.deleteObjects接口传入每个待删除对象的VersionId参数批量删除多版本对象，接口定义请参见[批量删除对象\(Node.js SDK\)](#)。

代码示例一：删除单个多版本对象

本示例用于删除examplebucket桶中的example/objectname对象，对应的版本号是G001117FCE89978B0000401205D5DC9A。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定删除对象，此处以 example/objectname 为例
      Key: 'example/objectname',
      // 指定版本对象的版本号
      VersionId: 'G001117FCE89978B0000401205D5DC9A'
    };
    // 删除版本对象
    const result = await obsClient.deleteObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

deleteObject();
```


代码示例二：批量删除多版本对象

本示例用于批量删除examplebucket桶中的objectname1、objectname2、objectname3对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteObjects() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定删除列表
      Objects: [
        { Key: 'objectname1', VersionId: "version1" },
        { Key: 'objectname2', VersionId: "version2" },
        { Key: 'objectname3', VersionId: "version3" }
      ]
    };
    // 批量删除对象
    const result = await obsClient.deleteObjects(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      // 获取删除成功的对象
      console.log("Deleted:");
      for (let i = 0; i < result.InterfaceResult.Deleted.length; i++) {
        const deleted = result.InterfaceResult.Deleted[i];
        console.log("Deleted[%d]-Key:%s, VersionId:%s", i, deleted.Key, deleted.VersionId);
      };
      // 获取删除失败的对象
      console.log("Errors:");
      for (let i = 0; i < result.InterfaceResult.Errors.length; i++) {
        const err = result.InterfaceResult.Errors[i];
        console.log("Errors[%d]-Key:%s, Code:%s", i, err.Key, err.Code);
      };
      console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
      console.log("Status: %d", result.CommonMsg.Status);
      console.log("Code: %s", result.CommonMsg.Code);
      console.log("Message: %s", result.CommonMsg.Message);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
    } catch (error) {
      console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
      console.log(error);
    };
  };
};

deleteObjects();
```

12 生命周期管理(Node.js SDK)

12.1 设置桶的生命周期配置(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

OBS支持用户配置指定的规则，实现定时删除桶中的对象或者定时转换对象的存储类别，从而节省存储费用，更多生命周期相关信息请参见[生命周期管理](#)。

调用设置桶的生命周期配置接口，您可以为指定桶设置生命周期策略。

说明

- 对象生命周期到期以后，对象将会永久删除，无法恢复。
- 归档存储不支持多AZ，因此不支持使用生命周期的存储类别转换功能，将多AZ的桶或对象转化为归档存储。
- 低频访问存储的最低存储时间为30天，归档存储的最低存储时间为90天。若对象经过转换，归档存储时间少于最低存储时间，需要补足剩余天数的存储费用。

接口约束

- 单个桶的生命周期规则条数没有限制，但一个桶中所有生命周期规则的XML描述总大小不能超过20KB。
- 您最多可以在一个并行文件系统下配置20条生命周期管理规则，超过20条将提示不支持。
- 您必须是桶拥有者或拥有设置桶的生命周期配置的权限，才能设置桶的生命周期配置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutLifecycleConfiguration权限，如果使用桶策略则需授予PutLifecycleConfiguration权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

ObsClient.setBucketLifecycle(params)

请求参数

表 12-1 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“-”和“.”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Rules	LifecycleRule []	必选	<p>参数解释: 桶生命周期规则列表。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 对象桶的生命周期规则条数没有限制，但一个桶中所有生命周期规则的XML描述总大小不能超过20KB。 您最多可以在一个并行文件系统下配置20条生命周期管理规则，超过20条将提示不支持。 <p>取值范围: 详见LifecycleRule。</p> <p>默认取值: 无</p>

表 12-2 LifecycleRule

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时可选	<p>参数解释: 生命周期规则ID。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过255的字符串。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Prefix	string	作为请求参数时必选	<p>参数解释: 对象名前缀，用以标识哪些对象可以匹配到当前这条规则。可为空字符串，代表匹配桶内所有对象。</p> <p>例如，假设您拥有以下对象：logs/day1、logs/day2、logs/day3和ExampleObject.jpg。如果您将ExampleObject.jpg指定为前缀，则规则仅应用于该特定对象。如果您将logs/指定为前缀，规则将应用于密钥名称以字符串“logs/”开头的三个对象。如果您指定空的前缀，规则将应用于桶中的所有对象。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
Status	string	作为请求参数时必选	<p>参数解释: 标识当前规则是否启用。</p> <p>约束限制: 无</p> <p>取值范围: Enabled: 启用 Disabled: 不启用。</p> <p>默认取值: 无</p>
Transitions	Transition []	作为请求参数时可选	<p>参数解释: 对象转换策略列表，用于使用生命周期规则转换对象的存储类别，包含生命周期配置的迁移时间和迁移后对象存储级别。</p> <p>约束限制: 仅针对对象的最新版本。</p> <p>取值范围: 详见Transition。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Expiration	Expiration	作为请求参数时可选	<p>参数解释: 对象过期时间配置。</p> <p>约束限制: 仅针对对象的最新版本。</p> <p>取值范围: 详见Expiration。</p> <p>默认取值: 无</p>
NoncurrentVersionTransitions	NoncurrentVersionTransition	作为请求参数时可选	<p>参数解释: 历史版本对象转换策略列表，用于使用生命周期规则转换历史版本对象的存储类别，包含生命周期配置的迁移时间和迁移后对象存储级别。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 仅针对对象的历史版本。 • 当前桶已启用（或开启后暂停）多版本。 <p>取值范围: 详见NoncurrentVersionTransition。</p> <p>默认取值: 无</p>
NoncurrentVersionExpiration	NoncurrentVersionExpiration	作为请求参数时可选	<p>参数解释: 历史版本对象过期时间配置。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 仅针对对象的历史版本。 • 当前桶已启用（或开启后暂停）多版本。 • 并行文件系统不支持该参数。 <p>取值范围: 详见NoncurrentVersionExpiration。</p> <p>默认取值: 无</p>

说明

Transitions、Expiration、NoncurrentVersionTransitions、NoncurrentVersionExpiration不能全为空。

表 12-3 Transition

参数名称	参数类型	是否必选	描述
StorageClasses	StorageClassType	作为请求参数时必选	<p>参数解释: 对象转换后的存储类型。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 不支持“标准存储”类型。 对象存储类别转换限制: <ul style="list-style-type: none"> 仅支持将标准存储对象转换为低频访问存储对象，低频访问存储对象转换为标准存储对象需手动转换。 仅支持将标准存储或低频访问存储对象转换为归档存储对象。如果要将归档存储对象转换为标准存储或低频访问存储对象，需要手动先恢复对象，然后手动转换存储类别。 归档存储不支持多AZ。因此不支持使用生命周期的存储类别转换功能，将多AZ桶或对象的存储类别转化为归档存储。 <p>取值范围: 生命周期转换的目标存储类别取值范围详见StorageClassType。</p> <p>默认取值: 无</p>
Date	string	作为请求参数时，如果没有设置Days则必选	<p>参数解释: 指定OBS对该日期之前的对象执行生命周期规则。</p> <p>约束限制: 日期格式必须为ISO8601的格式，并且为UTC的零点。例如： 2018-01-01T00:00:00.000Z，表示将最后修改时间早于2018-01-01T00:00:00.000Z的对象转换成其他存储类型，等于或晚于这个时间的对象不会被转换成其他存储类型。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
Days	number	作为请求参数时, 如果没有设置 Date 则必选	<p>参数解释: 指定生命周期规则在对象最后更新过后多少天生效。</p> <p>约束限制: 仅适用于对象的最新版本。</p> <p>取值范围: 0~ (2³¹-1), 单位: 天。</p> <p>默认取值: 无</p>

表 12-4 Expiration

参数名称	参数类型	是否必选	描述
Date	string	作为请求参数时, 如果没有设置 Days 则必选	<p>参数解释: 指定OBS对该日期之前的对象执行生命周期规则, 即执行过期删除操作。</p> <p>约束限制: 日期格式必须为ISO8601的格式, 并且为UTC的零点。例如: 2018-01-01T00:00:00.000Z, 表示将最后修改时间早于2018-01-01T00:00:00.000Z的对象删除, 等于或晚于这个时间的对象不会被删除。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
Days	number	作为请求参数时, 如果没有设置 Date 则必选	<p>参数解释: 指定生命周期规则在对象最后更新过后多少天生效, 即执行过期删除操作。</p> <p>约束限制: 仅适用于对象的最新版本。</p> <p>取值范围: 1~ (2³¹-1), 单位: 天</p> <p>默认取值: 无</p>

表 12-5 NoncurrentVersionTransition

参数名称	参数类型	描述	说明
StorageClass	StorageClassType	作为请求参数时必选	<p>参数解释: 历史版本对象转换后的存储类型。</p> <p>约束限制:</p> <ul style="list-style-type: none">不支持“标准存储”类型。对象存储类别转换限制:<ul style="list-style-type: none">仅支持将标准存储对象转换为低频访问存储对象，低频访问存储对象转换为标准存储对象需手动转换。仅支持将标准存储或低频访问存储对象转换为归档存储对象。如果要将归档存储对象转换为标准存储或低频访问存储对象，需要手动先恢复对象，然后手动转换存储类别。归档存储不支持多AZ。因此不支持使用生命周期的存储类别转换功能，将多AZ桶或对象的存储类别转化为归档存储。 <p>取值范围: 生命周期转换的目标存储类别取值范围详见StorageClassType。</p> <p>默认取值: 无</p>
NoncurrentDays	number	作为请求参数时必选	<p>参数解释: 表示对象在成为历史版本之后第几天时规则生效。</p> <p>约束限制: 仅适用于历史版本。</p> <p>取值范围: 0~(2³¹-1)，单位：天</p> <p>默认取值: 无</p>

表 12-6 NoncurrentVersionExpiration

参数名称	参数类型	是否必选	描述
NoncurrentDays	number	作为请求参数时必选	<p>参数解释: 表示对象在成为历史版本之后第几天时规则生效。</p> <p>约束限制: 仅针对历史版本。</p> <p>取值范围: 0~(2³¹-1)，单位：天</p> <p>默认取值: 无</p>

表 12-7 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	<p>标准存储。</p> <p>标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。</p>
ObsClient.enums.StorageClassWarm	WARM	<p>低频访问存储。</p> <p>低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。</p>
ObsClient.enums.StorageClassCold	COLD	<p>归档存储。</p> <p>归档存储适用于很少访问（平均一年访问一次）数据的业务场景。</p>

返回结果

表 12-8 返回结果

参数类型	描述
<p>表12-9</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表12-9。</p>

表 12-9 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表12-11	参数解释: 操作成功后的结果数据，详见 表12-11 。 约束限制: 当Status大于300时为空。

表 12-10 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 12-11 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例：设置对象转换策略

本示例用于设置名为examplebucket桶的生命周期规则，指定最新版本对象和历史版本对象的转换策略。

代码示例：设置对象过期时间

本示例用于设置名为examplebucket桶的生命周期规则，指定最新版本对象和历史版本对象的过期时间。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketLifecycle() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 生命周期配置
      Rules: [
        {
          ID:'rule1',Prefix:'prefix1',Status:'Enabled',
          // 指定满足前缀的对象创建60天后过期
          Expiration:{Days:60},
          // 指定满足前缀的对象成为历史版本60天后过期
          NoncurrentVersionExpiration:{NoncurrentDays : 60}
        },
        {
          ID:'rule2',Prefix:'prefix2',Status:'Enabled',
          // 直接指定满足前缀的对象过期时间，该值必须符合ISO8601格式，而且必须是UTC午夜0点
          Expiration:{Date: '2018-12-31T00:00:00Z'},
        }
      ]
    };
    // 设置桶的生命周期配置
    const result = await obsClient.setBucketLifecycle(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s lifecycle configuration successful!", params.Bucket);
    }
  }
}
```

```
console.log("RequestId: %s", result.CommonMsg.RequestId);
return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
console.log(error);
};
};
setBucketLifecycle();
```

12.2 获取桶的生命周期配置(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

OBS支持用户配置指定的规则，实现定时删除桶中的对象或者定时转换对象的存储类别，从而节省存储费用，更多生命周期相关信息请参见[生命周期管理](#)。

调用获取桶的生命周期配置接口，您可以获取指定桶的生命周期策略。

接口约束

- 您必须是桶拥有者或拥有获取桶的生命周期配置的权限，才能获取桶的生命周期配置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:GetLifecycleConfiguration权限，如果使用桶策略则需授予GetLifecycleConfiguration权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getBucketLifecycle(params)
```

请求参数

表 12-12 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 12-13 返回结果

参数类型	描述
<p>表12-14</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表12-14。</p>

表 12-14 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表12-16	参数解释: 操作成功后的结果数据，详见 表12-16 。 约束限制: 当Status大于300时为空。

表 12-15 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 12-16 GetBucketLifecycleOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。 默认取值: 无
Rules	LifecycleRule []	必选 参数解释: 桶生命周期规则列表, 详见 LifecycleRule 。

表 12-17 LifecycleRule

参数名称	参数类型	描述
ID	string	参数解释: 生命周期规则ID。
Prefix	string	参数解释: 对象名前缀, 用以标识哪些对象可以匹配到当前这条规则。可为空字符串, 代表匹配桶内所有对象。 例如, 假设您拥有以下对象: logs/day1、logs/day2、logs/day3和ExampleObject.jpg。如果您将ExampleObject.jpg指定为前缀, 则规则仅应用于该特定对象。如果您将logs/指定为前缀, 规则将应用于密钥名称以字符串“logs/”开头的三个对象。如果您指定空的前缀, 规则将应用于桶中的所有对象。
Status	string	参数解释: 标识当前规则是否启用。 取值范围: Enabled: 启用 Disabled: 不启用。
Transitions	Transition []	参数解释: 对象转换策略列表, 用于使用生命周期规则转换对象的存储类别, 包含生命周期配置的迁移时间和迁移后对象存储级别, 详见 Transition 。 约束限制: 仅针对对象的最新版本。

参数名称	参数类型	描述
Expiration	Expiration	<p>参数解释: 对象过期时间配置, 详见Expiration。</p> <p>约束限制: 仅针对对象的最新版本。</p>
NoncurrentVersionTransitions	NoncurrentVersionTransition []	<p>参数解释: 历史版本对象转换策略列表, 用于使用生命周期规则转换历史版本对象的存储类别, 包含生命周期配置的迁移时间和迁移后对象存储级别, 详见NoncurrentVersionTransition。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 仅针对对象的历史版本。 • 当前桶已启用 (或开启后暂停) 多版本。
NoncurrentVersionExpiration	NoncurrentVersionExpiration	<p>参数解释: 历史版本对象过期时间配置, 详见NoncurrentVersionExpiration。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 仅针对对象的历史版本。 • 当前桶已启用 (或开启后暂停) 多版本。 • 并行文件系统不支持该参数。

表 12-18 Transition

参数名称	参数类型	描述
StorageClass	StorageClassType	<p>参数解释: 对象转换后的存储类型。</p>
Date	string	<p>参数解释: 指定OBS对该日期之前的对象执行生命周期规则。</p>
Days	number	<p>参数解释: 指定生命周期规则在对象最后更新过后多少天生效。</p>

表 12-19 Expiration

参数名称	参数类型	描述
Date	string	参数解释: 指定OBS对该日期之前的对象执行生命周期规则，即执行过期删除操作。
Days	number	参数解释: 指定生命周期规则在对象最后更新过后多少天生效，即执行过期删除操作。

表 12-20 NoncurrentVersionTransition

参数名称	参数类型	描述
StorageClass	StorageClassType	参数解释: 历史版本对象转换后的存储类型。
NoncurrentDays	number	参数解释: 表示对象在成为历史版本之后第几天时规则生效。

表 12-21 NoncurrentVersionExpiration

参数名称	参数类型	描述
NoncurrentDays	number	参数解释: 表示对象在成为历史版本之后第几天时规则生效。

表 12-22 AbortIncompleteMultipartUpload

参数名称	参数类型	描述
DaysAfterInitiation	number	参数解释: 指定未合并的多段任务（碎片）的生命周期规则，在多段任务初始化过后多少天生效。

表 12-23 StorageClassType

常量名	原始值	说明
ObsClient.enums.StorageClassStandard	STANDARD	标准存储。 标准存储拥有低访问时延和较高的吞吐量，适用于有大量热点对象（平均一个月多次）或小对象（<1MB），且需要频繁访问数据的业务场景。
ObsClient.enums.StorageClassWarm	WARM	低频访问存储。 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求能够快速访问数据的业务场景。
ObsClient.enums.StorageClassCold	COLD	归档存储。 归档存储适用于很少访问（平均一年访问一次）数据的业务场景。

代码示例

您可以通过ObsClient.getBucketLifecycle查看桶的生命周期规则。以下代码展示了如何获取名为examplebucket桶的生命周期规则。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketLifecycle() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket"
    };
    // 获取桶的生命周期规则
    const result = await obsClient.getBucketLifecycle(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s lifecycle configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let i = 0; i < result.InterfaceResult.Rules.length; i++) {
        const rule = result.InterfaceResult.Rules[i];
        console.log("ID:%s, Prefix:%s, Status:%s, Expiration[Date]:%s, Expiration[Days]:%d, NoncurrentVersionExpiration[NoncurrentDays]:%s",
          rule.ID, rule.Prefix, rule.Status, rule.Expiration.Date,
          rule.Expiration.Days, rule.NoncurrentVersionExpiration.NoncurrentDays);
      }
    }
  } catch (error) {
    console.error(error);
  }
}
```

```
);  
};  
return;  
};  
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
console.log("Status: %d", result.CommonMsg.Status);  
console.log("Code: %s", result.CommonMsg.Code);  
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
console.log(error);  
};  
};  
};  
getBucketLifecycle();
```

12.3 删除桶的生命周期配置(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

OBS支持用户配置指定的规则，实现定时删除桶中的对象或者定时转换对象的存储类别，从而节省存储费用，更多生命周期相关信息请参见[生命周期管理](#)。

调用删除桶的生命周期配置接口，您可以删除指定桶的生命周期策略。

接口约束

- 您必须是桶拥有者或拥有删除桶的生命周期配置的权限，才能删除桶的生命周期配置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutLifecycleConfiguration权限，如果使用桶策略则需授予PutLifecycleConfiguration权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.deleteBucketLifecycle(params)
```

请求参数

表 12-24 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 12-25 返回结果

参数类型	描述
<p>表12-26</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表12-26。</p>

表 12-26 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表12-28	参数解释: 操作成功后的结果数据，详见 表12-28 。 约束限制: 当Status大于300时为空。

表 12-27 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 12-28 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

您可以通过ObsClient.deleteBucketLifecycle删除桶的生命周期规则。以下代码展示了如何删除名为examplebucket桶的生命周期规则。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucketLifecycle() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
    };
    // 删除桶的生命周期配置
    const result = await obsClient.deleteBucketLifecycle(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s lifecycle configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

deleteBucketLifecycle();
```

13 桶的 CORS 配置(Node.js SDK)

13.1 设置桶的 CORS 配置(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

跨域资源共享（Cross Origin Resource Sharing，CORS）是由W3C标准化组织提出的一种网络浏览器的规范机制，定义了一个域中加载的客户端Web应用程序与另一个域中的资源交互的方式。而在通常的网页请求中，由于同源安全策略（Same Origin Policy，SOP）的存在，不同域之间的网站脚本和内容是无法进行交互的。OBS支持CORS规范，允许跨域请求访问OBS中的资源。

调用设置桶的CORS配置接口，您可设置指定桶的跨域资源共享规则，以允许客户端浏览器进行跨域请求。如果原规则存在则覆盖原规则。

接口约束

- 您必须是桶所有者或拥有设置桶的CORS配置的权限，才能设置桶的CORS配置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutBucketCORS权限，如果使用桶策略则需授予PutBucketCORS权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.setBucketCors(params)
```


请求参数

表 13-1 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“-”和“.”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>
CorsRules	CorsRule []	必选	<p>参数解释: 桶的CORS规则列表。</p> <p>约束限制: 列表最多包含100个规则。</p> <p>取值范围: 详见CorsRule。</p> <p>默认取值: 无</p>

表 13-2 CorsRule

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时可选	参数解释: CORS规则ID 约束限制: 无 取值范围: 长度大于0且不超过255的字符串。 默认取值: 无
AllowedMethod	string[]	作为请求参数时必选	参数解释: 指定允许的跨域请求HTTP方法，即桶和对象的几种操作类型。 约束限制: 无 取值范围: 支持以下HTTP方法： <ul style="list-style-type: none">• GET• PUT• HEAD• POST• DELETE 默认取值: 无
AllowedOrigin	string[]	作为请求参数时必选	参数解释: 指定允许的跨域请求的来源，即允许来自该域名下的请求访问该桶。 约束限制: 表示域名的字符串，每个匹配规则允许使用最多一个“*”通配符。例如：https://*.vbs.example.com。 取值范围: 无 默认取值: 无

参数名称	参数类型	是否必选	描述
AllowedHeader	string[]	作为请求参数时可选	<p>参数解释: 指定允许的跨域请求的头域。只有匹配上允许的头域中的配置，才被视为是合法的 CORS 请求。</p> <p>约束限制: 最多可填写一个“*”通配符，不支持 &、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
MaxAgeSeconds	number	作为请求参数时可选	<p>参数解释: 请求来源的客户端可以对跨域请求返回结果的缓存时间。</p> <p>约束限制: 每个 CORSRule 可以包含至多一个 MaxAgeSeconds。</p> <p>取值范围: 0~(2³¹-1)，单位：秒。</p> <p>默认取值: 100</p>
ExposeHeader	string[]	作为请求参数时可选	<p>参数解释: CORS 规则允许响应中可返回的附加头域，给客户端提供额外的信息。默认情况下浏览器只能访问以下头域：Content-Length、Content-Type，如果需要访问其他头域，需要在附加头域中配置。</p> <p>约束限制: 不支持*、&、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

返回结果

表 13-3 返回结果

参数类型	描述
表13-4 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见表13-4。

表 13-4 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表13-6	参数解释： 操作成功后的结果数据，详见表13-6。 约束限制： 当Status大于300时为空。

表 13-5 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释： OBS服务端返回的HTTP状态码。 取值范围： 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 。
Code	string	参数解释： OBS服务端返回的错误码。
Message	string	参数解释： OBS服务端返回的错误描述。
HostId	string	参数解释： OBS服务端返回的请求服务端ID。
RequestId	string	参数解释： OBS服务端返回的请求ID。

Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 13-6 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

本示例用于设置名为examplebucket桶的CORS规则配置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketCors() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定Cors规则
      CorsRules: [
        {
          // 指定允许的跨域请求方法(GET/PUT/DELETE/POST/HEAD)
          AllowedMethod: ['GET', 'HEAD', 'PUT'],
          // 指定允许跨域请求的来源
          AllowedOrigin: ['http://www.a.com', 'http://www.b.com'],
          // 控制在OPTIONS预取指令中Access-Control-Request-Headers头中指定的header是否被允许使用
          AllowedHeader: ['x-obs-header'],
          // 指定允许用户从应用程序中访问的header
          ExposeHeader: ['x-obs-expose-header'],
          // 指定浏览器对特定资源的预取(OPTIONS)请求返回结果的缓存时间,单位为秒
          MaxAgeSeconds: 10
        }
      ]
    };
  }
};
// 设置桶的CORS配置
```

```
const result = await obsClient.setBucketCors(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Set bucket(%s) CORS configuration successful!", params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
setBucketCors();
```

13.2 获取桶的 CORS 配置(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

跨域资源共享（Cross Origin Resource Sharing，CORS）是由W3C标准化组织提出的一种网络浏览器的规范机制，定义了一个域中加载的客户端Web应用程序与另一个域中的资源交互的方式。而在通常的网页请求中，由于同源安全策略（Same Origin Policy，SOP）的存在，不同域之间的网站脚本和内容是无法进行交互的。OBS支持CORS规范，允许跨域请求访问OBS中的资源。

调用获取桶的CORS配置接口，您可获取指定桶的跨域资源共享规则。

接口约束

- 您必须是桶所有者或拥有获取桶的CORS配置的权限，才能获取桶的CORS配置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:GetBucketCORS权限，如果使用桶策略则需授予GetBucketCORS权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getBucketCors(params)
```

请求参数

表 13-7 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 13-8 返回结果

参数类型	描述
<p>表13-9</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表13-9。</p>

表 13-9 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	表13-11	参数解释: 操作成功后的结果数据, 详见 表13-11 。 约束限制: 当Status大于300时为空。

表 13-10 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 13-11 GetBucketCorsOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
CorsRules	CorsRule []	参数解释: 桶的CORS规则列表，详情参考 CorsRule 。

表 13-12 CorsRule

参数名称	参数类型	是否必选	描述
ID	string	作为请求参数时可选	参数解释: CORS规则ID 约束限制: 无 取值范围: 长度大于0且不超过255的字符串。 默认取值: 无
AllowedMethod	string []	作为请求参数时必选	参数解释: 指定允许的跨域请求HTTP方法，即桶和对象的几种操作类型。 约束限制: 无 取值范围: 支持以下HTTP方法： <ul style="list-style-type: none">• GET• PUT• HEAD• POST• DELETE 默认取值: 无

参数名称	参数类型	是否必选	描述
AllowedOrigin	string[]	作为请求参数时必选	<p>参数解释: 指定允许的跨域请求的来源，即允许来自该域名下的请求访问该桶。</p> <p>约束限制: 表示域名的字符串，每个匹配规则允许使用最多一个“*”通配符。例如：https://*.vbs.example.com。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
AllowedHeader	string[]	作为请求参数时可选	<p>参数解释: 指定允许的跨域请求的头域。只有匹配上允许的头域中的配置，才被视为是合法的CORS请求。</p> <p>约束限制: 最多可填写一个“*”通配符，不支持&、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
MaxAgeSeconds	number	作为请求参数时可选	<p>参数解释: 请求来源的客户端可以对跨域请求返回结果的缓存时间。</p> <p>约束限制: 每个CORSRule可以包含至多一个MaxAgeSeconds。</p> <p>取值范围: 0~(2³¹-1)，单位：秒。</p> <p>默认取值: 100</p>

参数名称	参数类型	是否必选	描述
ExposeHeader	string[]	作为请求参数时可选	<p>参数解释: CORS规则允许响应中可返回的附加头域，给客户端提供额外的信息。默认情况下浏览器只能访问以下头域：Content-Length、Content-Type，如果需要访问其他头域，需要在附加头域中配置。</p> <p>约束限制: 不支持*、&、:、<、空格以及中文字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

代码示例

本示例用于获取名为examplebucket桶的CORS规则配置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketCors() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
    };
    // 获取桶的CORS配置
    const result = await obsClient.getBucketCors(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s) CORS configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let k = 0; k < result.InterfaceResult.CorsRules.length; k++) {
        const corsRule = result.InterfaceResult.CorsRules[k];
        console.log("ID:%s, AllowedOrigin:%s, AllowedMethod:%s, AllowedHeader:%s, MaxAgeSeconds:%d, ExposeHeader:%s",
          corsRule.ID, corsRule.AllowedOrigin.join("|"), corsRule.AllowedMethod.join("|"),
          corsRule.AllowedHeader.join("|"),
          corsRule.MaxAgeSeconds, corsRule.ExposeHeader.join("|")
        );
      }
    }
  } catch (error) {
    console.error("Error: ", error);
  }
}
```

```
};  
return;  
};  
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
console.log("Status: %d", result.CommonMsg.Status);  
console.log("Code: %s", result.CommonMsg.Code);  
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
  console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
  console.log(error);  
};  
};  
getBucketCors();
```

13.3 删除桶的 CORS 配置(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

跨域资源共享（Cross Origin Resource Sharing，CORS）是由W3C标准化组织提出的一种网络浏览器的规范机制，定义了一个域中加载的客户端Web应用程序与另一个域中的资源交互的方式。而在通常的网页请求中，由于同源安全策略（Same Origin Policy，SOP）的存在，不同域之间的网站脚本和内容是无法进行交互的。OBS支持CORS规范，允许跨域请求访问OBS中的资源。

调用删除桶的CORS配置接口，您可删除指定桶的跨域资源共享规则。

接口约束

- 您必须是桶拥有者或拥有删除桶的CORS配置的权限，才能删除桶的CORS配置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutBucketCORS权限，如果使用桶策略则需授予PutBucketCORS权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.deleteBucketCors(params)
```

请求参数

表 13-13 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

返回结果

表 13-14 返回结果

参数类型	描述
<p>表13-15</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表13-15。</p>

表 13-15 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息, 包含HTTP状态码, 操作失败的错误码等, 详见 ICommonMsg 。
InterfaceResult	表13-17	参数解释: 操作成功后的结果数据, 详见 表13-17 。 约束限制: 当Status大于300时为空。

表 13-16 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 13-17 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

以下代码展示了如何删除跨域规则:

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucketCors() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket"
    };
    // 删除桶的跨域规则
    const result = await obsClient.DeleteBucketCors(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s CORS configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("Delete bucket(%s)'s CORS configuration successful!", params.Bucket);
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

deleteBucketCors();
```

14 设置访问日志(Node.js SDK)

14.1 设置桶日志管理配置(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

说明

如果桶的存储类别为低频访问存储或归档存储，则不能作为日志目标桶。

功能说明

调用设置桶日志管理配置接口，您可以为指定桶打开桶日志功能，并配置日志存放的目标桶。创建桶时，默认是不生成桶的日志的，调用Node.js SDK的设置桶日志管理配置接口时，接口会自动为您打开桶日志配置。桶日志功能开启后，桶的每次操作将会产生一条日志，并将多条日志打包成一个日志文件。日志文件存放位置需要在开启桶日志功能时指定，可以存放到开启日志功能的桶中，也可以存放到其他您有权限的桶中，但需要和开启日志功能的桶在同一个region中。您还可以根据需要配置日志文件的访问权限，以及日志文件的文件名前缀。

接口约束

- 由于日志文件是OBS产生，并且由OBS上传到存放日志的桶中，因此OBS需要获得委托授权，用于上传生成的日志文件，所以在配置桶日志管理前，需要先到统一身份认证服务生成一个对OBS服务的委托，委托配置权限只需设置目标桶的上传对象权限。如何创建委托请参考[创建云服务委托](#)。
- 您必须是桶拥有者或拥有设置桶日志管理配置的权限，才能设置桶日志管理配置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutBucketLogging权限，如果使用桶策略则需授予PutBucketLogging权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

- 日志目标桶与源桶必须在同一个区域（region）。
- 在配置桶日志管理前，需要先到[统一身份认证服务](#)生成一个对OBS服务的委托，获取到委托名。请参考[创建IAM委托](#)。

方法定义

ObsClient.setBucketLogging(params)

请求参数

表 14-1 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。• 桶命名规则如下：<ul style="list-style-type: none">- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。- 禁止使用IP地址。- 禁止以“-”或“.”开头及结尾。- 禁止两个“.”相邻（如：“my..bucket”）。- 禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。• 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

参数名称	参数类型	是否必选	描述
Agency	string	作为请求参数时如果是设置桶日志配置则必选	<p>参数解释: 目标桶（存放日志文件的桶）的所有者通过统一身份认证服务（IAM）创建对OBS云服务委托的名称。</p> <p>约束限制: 默认情况下，在为委托配置权限时只需设置日志存储桶的上传对象（PutObject）权限。如果日志存储桶开启了默认加密功能，还需要委托同时具有日志存储桶所在区域的KMS Administrator权限。</p> <p>取值范围: 可以选择账号下已有的IAM委托，也可以创建一个新的委托。创建委托参见创建IAM委托。</p> <p>默认取值: 无</p>
LoggingEnabled	Logging Enabled	设置logging时必选。 关闭logging时勿选。	<p>参数解释: 日志配置信息。不配置时默认关闭桶日志。</p> <p>约束限制: 无</p> <p>取值范围: 详见LoggingEnabled。</p> <p>默认取值: 无</p>

表 14-2 LoggingEnabled

参数名称	参数类型	是否必选	描述
TargetBucket	string	设置 logging 时必选。 关闭 logging 时勿选。	参数解释: 日志存储桶的桶名, 该桶用于存放日志文件。 约束限制: <ul style="list-style-type: none">需要和开启日志功能的桶在同一个 region 中。桶的名字需全局唯一, 不能与已有的任何桶名称重复, 包括其他用户创建的桶。桶命名规则如下:<ul style="list-style-type: none">3~63 个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。禁止使用 IP 地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻(如: “my..bucket”)。禁止“.”和“-”相邻(如: “my-.bucket”和“my.-bucket”)。同一用户在同一区域多次创建同名桶不会报错, 创建的桶属性以第一次请求为准。 取值范围: 长度为3~63个字符。 默认取值: 无
TargetPrefix	string	设置 logging 时必选。 关闭 logging 时勿选。	参数解释: 日志文件对象的对象名前缀, 设置后日志文件对象将携带统一前缀存储在日志存储桶中。 约束限制: 无 取值范围: 长度大于0且不超过1024的字符串。 默认取值: 无

参数名称	参数类型	是否必选	描述
TargetGrants	Grant	可选	<p>参数解释: 被授权用户的权限信息列表，定义哪些用户可以针对日志文件进行哪些访问操作。</p> <p>约束限制: 无</p> <p>取值范围: 详见Grant。</p> <p>默认取值: 无</p>

表 14-3 Grant

参数名称	参数类型	是否必选	描述
Grantee	Grantee	必选	<p>参数解释: 被授权用户相关信息。</p> <p>约束限制: 无</p> <p>取值范围: 详见Grantee。</p> <p>默认取值: 无</p>
Permission	PermissionType	必选	<p>参数解释: 被授予的权限。</p> <p>约束限制: 无</p> <p>取值范围: 详见PermissionType。</p> <p>默认取值: 无</p>

表 14-4 Grantee

参数名称	参数类型	是否必选	描述
Type	GranteeType	作为请求参数时必选	参数解释: 被授权用户的类型。 约束限制: 无 取值范围: 被授权用户类型的取值范围详见 GranteeType 。 默认取值: 无
ID	string	作为请求参数时, 如果Type为用户类型则必选。	参数解释: 被授权用户的账号ID, 即domain_id。 约束限制: 无 取值范围: 如何获取账号ID请参见 获取账号ID 。 默认取值: 无
Name	string	作为请求参数时可选	参数解释: 被授权用户的账号名。 约束限制: <ul style="list-style-type: none">只能以字母开头。长度为6-32个字符。只能包含英文字母、数字或特殊字符(-_)。 取值范围: 无 默认取值: 无
URI	GroupUriType	作为请求参数时, 如果Type为用户组类型则必选。	参数解释: 被授权的用户组。 约束限制: 无 取值范围: 授权用户组取值范围详见 GroupUriType 。 默认取值: 无

表 14-5 GranteeType

常量名	原始值	说明
GranteeGroup	Group	授权给用户组。
GranteeUser	CanonicalUser	授权给单个用户。

表 14-6 GroupUriType

常量名	原始值	说明
GroupAllUsers	AllUsers	所有用户。
GroupAuthenticatedUsers	AuthenticatedUsers	授权用户，已废弃。
GroupLogDelivery	LogDelivery	日志投递组，已废弃。

表 14-7 PermissionType

常量名	原始值	说明
ObsClient.enums.PermissionRead	READ	如果有桶的读权限，则可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 如果有对象的读权限，则可以获取该对象内容和元数据。
ObsClient.enums.PermissionWrite	WRITE	如果有桶的写权限，则可以上传、覆盖和删除该桶内任何对象和段。 此权限在对象上不适用。
ObsClient.enums.PermissionReadAcp	READ_ACP	如果有读ACP的权限，则可以获取对应的桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有读对应桶或对象ACP的权限。
ObsClient.enums.PermissionWriteAcp	WRITE_ACP	如果有写ACP的权限，则可以更新对应桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。 拥有了写ACP的权限，由于可以更改权限控制策略，实际上意味着拥有了完全访问的权限。
ObsClient.enums.PermissionFullControl	FULL_CONTROL	如果有桶的完全控制权限意味着拥有PermissionRead、PermissionWrite、PermissionReadAcp和PermissionWriteAcp的权限。 如果有对象的完全控制权限意味着拥有PermissionRead、PermissionReadAcp和PermissionWriteAcp的权限。

返回结果

表 14-8 返回结果

参数类型	描述
表14-9 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见 表 14-9 。

表 14-9 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表14-11	参数解释： 操作成功后的结果数据，详见 表14-11 。 约束限制： 当Status大于300时为空。

表 14-10 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释： OBS服务端返回的HTTP状态码。 取值范围： 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 。
Code	string	参数解释： OBS服务端返回的错误码。
Message	string	参数解释： OBS服务端返回的错误描述。
HostId	string	参数解释： OBS服务端返回的请求服务端ID。
RequestId	string	参数解释： OBS服务端返回的请求ID。

Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 14-11 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例一：开启桶日志

本示例用于设置名为examplebucket桶的日志管理配置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketLogging() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定委托的名称, 此处以 obs_test_agency 为例
      Agency: 'obs_test_agency',
      LoggingEnabled: {
        // 指定用于存放产生的日志文件的桶名, 此处以 TargetBucketname 为例
        TargetBucket: 'TargetBucketname',
        // 指定用于生成的日志文件的名称前缀, 此处以 TargetPrefixtest/ 为例
        TargetPrefix: 'TargetPrefixtest/',
        // 指定被授权用户权限信息列表
        TargetGrants: [
          // 指定特定用户拥有读的权限, 此处用户ID为0a03f5833900d3730f13c00f49d5exxx
          { Grantee: { Type: 'CanonicalUser', ID: '0a03f5833900d3730f13c00f49d5exxx' }, Permission:
            obsClient.enums.PermissionRead }
        ]
      }
    };
    // 设置桶的访问日志配置
    const result = await obsClient.setBucketLogging(params);
  }
}
```



```
        if (result.CommonMsg.Status <= 300) {
            console.log("Set bucket(%s)'s logging configuration successful!", params.Bucket);
            console.log("RequestId: %s", result.CommonMsg.RequestId);
            return;
        };
        console.log("An ObsError was found, which means your request sent to OBS was rejected with an
error response.");
        console.log("Status: %d", result.CommonMsg.Status);
        console.log("Code: %s", result.CommonMsg.Code);
        console.log("Message: %s", result.CommonMsg.Message);
        console.log("RequestId: %s", result.CommonMsg.RequestId);
    } catch (error) {
        console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
        console.log(error);
    };
};

setBucketLogging();
```

代码示例二：关闭桶日志

本示例用于关闭名为examplebucket桶的日志管理配置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
    // 推荐通过环境变量获取AK/SK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
    access_key_id: process.env.ACCESS_KEY_ID,
    secret_access_key: process.env.SECRET_ACCESS_KEY,
    // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露
    // 风险。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
    // security_token: process.env.SECURITY_TOKEN,
    // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
    server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketLogging() {
    try {
        const params = {
            // 指定存储桶名称
            Bucket: "examplebucket",
            // 清空日志配置
            LoggingEnabled : {}
        };
        // 删除桶的访问日志配置
        const result = await obsClient.setBucketLogging(params);
        if (result.CommonMsg.Status <= 300) {
            console.log("Delete bucket(%s)'s logging configuration successful!", params.Bucket);
            console.log("RequestId: %s", result.CommonMsg.RequestId);
            return;
        };
        console.log("An ObsError was found, which means your request sent to OBS was rejected with an
error response.");
        console.log("Status: %d", result.CommonMsg.Status);
        console.log("Code: %s", result.CommonMsg.Code);
        console.log("Message: %s", result.CommonMsg.Message);
        console.log("RequestId: %s", result.CommonMsg.RequestId);
    } catch (error) {
        console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
        console.log(error);
    }
}
```

```
};  
};  
setBucketLogging();
```

14.2 获取桶日志管理配置(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

调用获取桶日志管理配置，可获取指定桶的日志配置。

接口约束

- 您必须是桶所有者或拥有获取桶日志管理配置的权限，才能获取桶日志管理配置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:GetBucketLogging权限，如果使用桶策略则需授予GetBucketLogging权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getBucketLogging(params)
```

请求参数

表 14-12 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 14-13 返回结果

参数类型	描述
<p>表14-14</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表14-14。</p>

表 14-14 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表14-16	参数解释: 操作成功后的结果数据，详见 表14-16 。 约束限制: 当Status大于300时为空。

表 14-15 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 14-16 GetBucketLoggingOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
Agency	string	参数解释: 目标桶（存放日志文件的桶）的所有者通过统一身份认证服务（IAM）创建对OBS云服务委托的名称。 可以选择账号下已有的IAM委托，也可以创建一个新的委托。创建委托参见 创建IAM委托 。
LoggingEnabled	LoggingEnabled	参数解释: 日志配置信息。

表 14-17 LoggingEnabled

参数名称	参数类型	描述
TargetBucket	string	参数解释: 日志存储桶的桶名，该桶用于存放日志文件。
TargetPrefix	string	参数解释: 日志文件对象的对象名前缀，设置后日志文件对象将携带统一前缀存储在日志存储桶中。
TargetGrants	Grant	参数解释: 被授权用户的权限信息列表，定义哪些用户可以针对日志文件进行哪些访问操作，详见 Grant 。

表 14-18 Grant

参数名称	参数类型	是否必选	描述
Grantee	Grantee	作为请求参数时必选	参数解释: 被授权用户相关信息，详见 Grantee 。
Permission	PermissionType	作为请求参数时必选	参数解释: 被授予的权限，详见 PermissionType 。

表 14-19 Grantee

参数名称	参数类型	是否必选	描述
Type	GranteeType	作为请求参数时必选	参数解释: 被授权用户的类型, 详见 GranteeType 。
ID	string	作为请求参数时, 如果Type为用户类型则必选。	参数解释: 被授权用户的账号ID, 即domain_id。
Name	string	作为请求参数时可选	参数解释: 被授权用户的账号名。
URI	GroupUriType	作为请求参数时, 如果Type为用户组类型则必选。	参数解释: 被授权的用户组, 详见 GroupUriType 。

表 14-20 GranteeType

常量名	原始值	说明
GranteeGroup	Group	授权给用户组。
GranteeUser	CanonicalUser	授权给单个用户。

表 14-21 GroupUriType

常量名	原始值	说明
GroupAllUsers	AllUsers	所有用户。
GroupAuthenticatedUsers	AuthenticatedUsers	授权用户, 已废弃。
GroupLogDelivery	LogDelivery	日志投递组, 已废弃。

表 14-22 PermissionType

常量名	原始值	说明
ObsClient.enums.PermissionRead	READ	如果有桶的读权限, 则可以获取该桶内对象列表、桶内多段任务、桶的元数据、桶的多版本。 如果有对象的读权限, 则可以获取该对象内容和元数据。

常量名	原始值	说明
ObsClient.enums.PermissionWrite	WRITE	如果有桶的写权限，则可以上传、覆盖和删除该桶内任何对象和段。 此权限在对象上不适用。
ObsClient.enums.PermissionReadACP	READ_ACP	如果有读ACP的权限，则可以获取对应的桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有读对应桶或对象ACP的权限。
ObsClient.enums.PermissionWriteACP	WRITE_ACP	如果有写ACP的权限，则可以更新对应桶或对象的权限控制列表（ACL）。 桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。 拥有了写ACP的权限，由于可以更改权限控制策略，实际上意味着拥有了完全访问的权限。
ObsClient.enums.PermissionFullControl	FULL_CONTROL	如果有桶的完全控制权限意味着拥有PermissionRead、PermissionWrite、PermissionReadAcp和PermissionWriteAcp的权限。 如果有对象的完全控制权限意味着拥有PermissionRead、PermissionReadAcp和PermissionWriteAcp的权限。

代码示例

您可以通过ObsClient.getBucketLogging获取桶日志配置。以下代码展示了如何获取名为examplebucket桶的日志管理配置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketLogging() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
    };
    // 设置桶的访问日志配置
    const result = await obsClient.getBucketLogging(params);
  }
}
```

```
if (result.CommonMsg.Status <= 300) {
  console.log("Get bucket(%s)'s logging configuration successful!", params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  if (result.InterfaceResult.LoggingEnabled) {
    console.log("TargetBucket: %s", result.InterfaceResult.LoggingEnabled.TargetBucket);
    console.log("TargetPrefix: %s", result.InterfaceResult.LoggingEnabled.TargetPrefix);
    for (let i = 0; i < result.InterfaceResult.LoggingEnabled.TargetGrants.length; i++) {
      const grant = result.InterfaceResult.LoggingEnabled.TargetGrants[0];
      console.log('Grant[%d]-Type:%s, ID:%s, URI:%s, Permission:%s',
        i, grant.Grantee.Type, grant.Grantee.ID, grant.Grantee.URI, grant.Permission
      );
    };
  };
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an
error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
getBucketLogging();
```


15 静态网站托管(Node.js SDK)

15.1 静态网站托管简介(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

您可以将静态网站文件上传至OBS的桶中作为对象，并对这些对象赋予公共读权限，然后将该桶配置成静态网站托管模式，以实现在OBS上托管静态网站的目的。第三方用户在访问您网站的时候，实际上是在访问OBS的桶中的对象。在使用静态网站托管功能时，OBS还支持配置请求重定向，通过重定向配置您可以将特定的请求或所有请求实施重定向。

更多关于静态网站托管的内容请参考[静态网站托管](#)。

您可通过以下步骤实现网站文件托管：

- 步骤1** 将网站文件上传至OBS的桶中，并设置对象MIME类型。
- 步骤2** 设置对象访问权限为公共读。
- 步骤3** 通过浏览器访问对象。

----结束

以下代码展示了如何实现网站文件托管：

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  //推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/usermanual-ca/ca_01_0003.htmlhttps://support.huaweicloud.com/intl/zh-cn/usermanual-ca/
```

```
ca_01_0003.htmlhttps://support.huaweicloud.com/eu/usermanual-ca/ca_01_0003.html
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露
// 风险。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。
// security_token: process.env.SECURITY_TOKEN,
// endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写。
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象，此处以 test.html 为例。
      Key: 'test.html',
      Body: '<html><header></header><body><h1>Hello OBS</h1></body></html>',
      // 设置对象MIME类型
      ContentType: 'text/html',
      // 设置对象访问权限为公共读
      ACL: obsClient.enums.AclPublicRead
    };

    // 上传对象
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put object(%s) under the bucket(%s) successful!!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageClass:%s, ETag:%s", result.InterfaceResult.StorageClass, result.InterfaceResult.ETag);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
    response.")
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
    attempting to communicate with OBS, for example, the client was unable to access the network.")
    console.log(error);
  }
}

putObject();
```

📖 说明

- 上例中您可以使用<http://examplebucket.your-endpoint/test.html>在浏览器直接访问托管的文件。
- 不支持在线预览，如果需要在线预览，请参考：https://support.huaweicloud.com/usermanual-obs/obs_03_0342.html

15.2 设置托管配置(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

OBS允许在桶内保存静态的网页资源，如.html网页文件、flash文件、音视频文件等，当客户端通过桶的Website接入点访问这些对象资源时，浏览器可以直接解析出这些支持的网页资源，呈现给最终用户。典型的应用场景有：

- 重定向所有的请求到另外一个站点。
- 设定特定的重定向规则来重定向特定的请求。

调用设置桶的网站配置接口，您可以为指定桶设置网站配置信息。

接口约束

- 尽量避免目标桶名中带有“.”，否则通过HTTPS访问时可能出现客户端校验证书出错。
- 设置桶的网站配置请求消息体的上限是10KB。
- 您必须是桶拥有者或拥有设置桶的网站配置的权限，才能设置桶的网站配置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutBucketWebsite权限，如果使用桶策略则需授予PutBucketWebsite权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.setBucketWebsite(params)
```

请求参数

表 15-1 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my.bucket”）。 禁止“.”和“-”相邻（如：“my.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>
RedirectAllRequestsTo	RedirectAllRequestTo	可选	<p>参数解释: 所有请求重定向规则。</p> <p>约束限制: 无</p> <p>取值范围: 详见RedirectAllRequestTo。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
IndexDocument	IndexDocument	可选	<p>参数解释: 默认页面配置。</p> <p>约束限制: 无</p> <p>取值范围: 详见IndexDocument。</p> <p>默认取值: 无</p>
ErrorDocument	ErrorDocument	可选	<p>参数解释: 错误页面配置。</p> <p>约束限制: 无</p> <p>取值范围: 详见ErrorDocument。</p> <p>默认取值: 无</p>
RoutingRules	RoutingRule[]	可选	<p>参数解释: 请求重定向规则列表。</p> <p>约束限制: 无</p> <p>取值范围: 详见RoutingRule。</p> <p>默认取值: 无</p>

说明

- ErrorDocument、IndexDocument和RoutingRules必须配套使用，且与RedirectAllRequestsTo互斥。当设置了这三个字段时，不能设置RedirectAllRequestsTo；反之，当设置了RedirectAllRequestsTo时，不能设置ErrorDocument、IndexDocument和RoutingRules。
- 当ErrorDocument、IndexDocument和RoutingRules三个字段一起使用时，RoutingRules可为空。
- ErrorDocument、IndexDocument、RoutingRules与RedirectAllRequestsTo不能全为空。

表 15-2 RedirectAllRequestsTo

参数名称	参数类型	是否必选	描述
HostName	string	如果 RedirectAllRequestsTo 设置必选	<p>参数解释: 重定向时使用的域名。如 www.example.com。</p> <p>约束限制: 域名需符合域名规范。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
Protocol	string	可选	<p>参数解释: 重定向请求时使用的协议。</p> <p>约束限制: 无</p> <p>取值范围: 协议类型可选: http、https。</p> <p>默认取值: 无</p>

表 15-3 IndexDocument

参数名称	参数类型	是否必选	描述
Suffix	string	如果 IndexDocument 设置则参数时必选	<p>参数解释: 该字段被追加在对文件夹的请求的末尾（例如: Suffix 参数设置为 “index.html”，请求的是 “samplebucket/images/”，返回的数据将是 “samplebucket” 桶内名为 “images/index.html” 的对象的内容）。</p> <p>约束限制: 该字段不能为空或者包含 “/” 字符。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>

表 15-4 ErrorDocument

参数名称	参数类型	是否必选	描述
Key	string	如果 ErrorDocument 设置时可选	<p>参数解释: 当4XX错误出现时使用的对象的名称。这个元素指定当错误出现时返回的页面。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>

表 15-5 RoutingRule

参数名称	参数类型	是否必选	描述
Condition	Condition	可选	<p>参数解释: 重定向规则生效需要满足的条件。</p> <p>约束限制: 无</p> <p>取值范围: 匹配条件选择范围详见Condition。</p> <p>默认取值: 无</p>
Redirect	Redirect	如果 RoutingRule 设置则参数时必选	<p>参数解释: 重定向请求时的具体信息。</p> <p>约束限制: 无</p> <p>取值范围: 详见Redirect。</p> <p>默认取值: 无</p>

表 15-6 Condition

参数名称	参数类型	是否必选	描述
KeyPrefixEquals	string	可选	<p>参数解释: 重定向生效时的对象名前缀。当向对象发送请求时, 如果对象名前缀等于这个值, 那么重定向生效。</p> <p>例如: 重定向ExamplePage.html对象的请求, KeyPrefixEquals设为ExamplePage.html。</p> <p>约束限制: 与HttpErrorcodeReturnedEquals参数不可同时使用, 两者二选一。</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
HttpErrorcodeReturnedEquals	string	可选	<p>参数解释: 重定向生效时的HTTP错误码。当发生错误时, 如果错误码等于这个值, 那么重定向生效。</p> <p>例如: 当返回的HTTP错误码为404时重定向到NotFound.html, 可以将Condition中的HttpErrorcodeReturnedEquals设置为404, Redirect中的ReplaceKeyWith设置为NotFound.html。</p> <p>约束限制: 与KeyPrefixEquals参数不可同时使用, 两者二选一。</p> <p>取值范围: 取值范围可参见错误码。</p> <p>默认取值: 无</p>

表 15-7 Redirect

参数名称	参数类型	是否必选	描述
Protocol	string	作为请求参数时可选	<p>参数解释: 重定向请求时使用的协议。</p> <p>约束限制: 无</p> <p>取值范围: 协议类型可选范围: http、https。</p> <p>默认取值: 无</p>
HostName	string	作为请求参数时可选	<p>参数解释: 重定向请求时使用的域名。</p> <p>约束限制: 域名需符合域名规范。</p> <p>取值范围: 无</p> <p>默认取值: 无</p>
ReplaceKeyPrefixWith	string	作为请求参数时可选	<p>参数解释: 重定向请求时使用的对象名前缀。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
ReplaceKeyWith	string	作为请求参数时可选	<p>参数解释: 重定向请求时使用的对象名。</p> <p>约束限制: 不可与replaceKeyPrefixWith同时使用。</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>

参数名称	参数类型	是否必选	描述
HttpRedirectCode	string	作为请求参数时可选	<p>参数解释: 重定向请求时响应中的HTTP状态码。</p> <p>约束限制: 无</p> <p>取值范围: 详见状态码。</p> <p>默认取值: 无</p>

返回结果

表 15-8 返回结果

参数类型	描述
<p>表15-9 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表15-9。</p>

表 15-9 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	<p>参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见ICommonMsg。</p>
InterfaceResult	表15-11	<p>参数解释: 操作成功后的结果数据，详见表15-11。</p> <p>约束限制: 当Status大于300时为空。</p>

表 15-10 ICommonMsg

参数名称	参数类型	描述
------	------	----

Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 15-11 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

本示例用于设置名为examplebucket桶的网站配置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。
  // 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。
  // security_token: process.env.SECURITY_TOKEN,
```

```
// endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写。
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketWebsite() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定默认页面配置, 此处以index.html为例
      IndexDocument: { Suffix: 'index.html' },
      // 指定错误页面配置, 此处以error.html为例
      ErrorDocument: { Key: 'error.html' },
      // 指定所有请求重定向
      RedirectAllRequestsTo : { HostName : 'www.example.com', Protocol : 'http' }
      // 指定请求重定向规则列表
      RoutingRules: [
        { Redirect: { HostName: "www.a.com", Protocol: obs.ProtocolHttp, ReplaceKeyPrefixWith: "prefix",
          HttpRedirectCode: "304" } },
        { Redirect: { HostName: "www.b.com", Protocol: obs.ProtocolHttps, ReplaceKeyWith: "replaceKey",
          HttpRedirectCode: "304" } },
      ]
    };
    // 设置桶的Website配置
    const result = await obsClient.setBucketWebsite(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s website configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

setBucketWebsite();
```

15.3 获取托管配置(Node.js SDK)

须知

开发过程中, 您有任何问题可以在github上[提交issue](#)。

功能介绍

OBS允许在桶内保存静态的网页资源, 如.html网页文件、flash文件、音视频文件等, 当客户端通过桶的Website接入点访问这些对象资源时, 浏览器可以直接解析出这些支持的网页资源, 呈现给最终用户。典型的应用场景有:

- 重定向所有的请求到另外一个站点。
- 设定特定的重定向规则来重定向特定的请求。

调用获取桶的网站配置接口, 您可以获取指定桶的网站配置信息。

接口约束

- 您必须是桶拥有者或拥有获取桶的网站配置的权限，才能获取桶的网站配置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:GetBucketWebsite权限，如果使用桶策略则需授予GetBucketWebsite权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

ObsClient.getBucketWebsite(params)

请求参数

表 15-12 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none">• 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。• 桶命名规则如下：<ul style="list-style-type: none">- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。- 禁止使用IP地址。- 禁止以“-”或“.”开头及结尾。- 禁止两个“.”相邻（如：“my..bucket”）。- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。• 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 15-13 返回结果

参数类型	描述
表15-14 说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。	参数解释： 接口返回信息，详见 表 15-14 。

表 15-14 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释： 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表15-16	参数解释： 操作成功后的结果数据，详见 表15-16 。 约束限制： 当Status大于300时为空。

表 15-15 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释： OBS服务端返回的HTTP状态码。 取值范围： 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 。
Code	string	参数解释： OBS服务端返回的错误码。
Message	string	参数解释： OBS服务端返回的错误描述。
HostId	string	参数解释： OBS服务端返回的请求服务端ID。
RequestId	string	参数解释： OBS服务端返回的请求ID。

Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 15-16 GetBucketWebsiteOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
RedirectAllRequestsTo	RedirectAllRequestTo	参数解释: 所有请求重定向规则，详见 RedirectAllRequestTo 。
IndexDocument	IndexDocument	参数解释: 默认页面配置，详见 IndexDocument 。
ErrorDocument	ErrorDocument	参数解释: 错误页面配置，详见 ErrorDocument 。
RoutingRules	RoutingRule []	参数解释: 请求重定向规则列表，详见 RoutingRule 。

表 15-17 IndexDocument

参数名称	参数类型	是否必选	描述
Suffix	string	如果 Index Document 设置则参数时必选	参数解释: 该字段被追加在对文件夹的请求的末尾（例如：Suffix参数设置为“index.html”，请求的是“samplebucket/images/”，返回的数据将是“samplebucket”桶内名为“images/index.html”的对象的内容）。 约束限制: 该字段不能为空或者包含“/”字符。 取值范围: 无 默认取值: 无

表 15-18 ErrorDocument

参数名称	参数类型	是否必选	描述
Key	string	如果 ErrorDocument 设置时可选	<p>参数解释: 当4XX错误出现时使用的对象的名称。这个元素指定当错误出现时返回的页面。</p> <p>约束限制: 无</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>

表 15-19 RedirectAllRequestsTo

参数名称	参数类型	描述
HostName	string	<p>参数解释: 重定向时使用的域名。如 www.example.com。</p>
Protocol	string	<p>参数解释: 重定向请求时使用的协议。</p>

表 15-20 RoutingRule

参数名称	参数类型	描述
Condition	Condition	<p>参数解释: 重定向规则生效需要满足的条件，详见 Condition。</p>
Redirect	Redirect	<p>参数解释: 重定向请求时的具体信息，详见 Redirect。</p>

表 15-21 Condition

参数名称	参数类型	是否必选	描述
KeyPrefixEquals	string	可选	<p>参数解释: 重定向生效时的对象名前缀。当向对象发送请求时, 如果对象名前缀等于这个值, 那么重定向生效。</p> <p>例如: 重定向ExamplePage.html对象的请求, KeyPrefixEquals设为ExamplePage.html。</p> <p>约束限制: 与HttpErrorCodeReturnedEquals参数不可同时使用, 两者二选一。</p> <p>取值范围: 长度大于0且不超过1024的字符串。</p> <p>默认取值: 无</p>
HttpErrorCodeReturnedEquals	string	可选	<p>参数解释: 重定向生效时的HTTP错误码。当发生错误时, 如果错误码等于这个值, 那么重定向生效。</p> <p>例如: 当返回的HTTP错误码为404时重定向到NotFound.html, 可以将Condition中的HttpErrorCodeReturnedEquals设置为404, Redirect中的ReplaceKeyWith设置为NotFound.html。</p> <p>约束限制: 与KeyPrefixEquals参数不可同时使用, 两者二选一。</p> <p>取值范围: 取值范围可参见错误码。</p> <p>默认取值: 无</p>

表 15-22 Redirect

参数名称	参数类型	描述
Protocol	string	<p>参数解释: 重定向请求时使用的协议。</p>
HostName	string	<p>参数解释: 重定向请求时使用的域名。</p>

参数名称	参数类型	描述
ReplaceKeyPrefixWith	string	参数解释: 重定向请求时使用的对象名前缀。
ReplaceKeyWith	string	参数解释: 重定向请求时使用的对象名。
HttpRedirectCode	string	参数解释: 重定向请求时响应中的HTTP状态码，详见 状态码 。

代码示例

您可以通过ObsClient.getBucketWebsite获取桶的托管配置。以下代码展示了如何获取名为examplebucket桶的网站配置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AK/SK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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。
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写。
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketWebsite() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
    };
    // 获取桶的网站配置
    const result = await obsClient.getBucketWebsite(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s website configuration successful!", params.Bucket);
      console.log('RedirectAllRequestsTo:');
      console.log('HostName: %s', result.InterfaceResult.RedirectAllRequestsTo.HostName);
      console.log('Protocol: %s', result.InterfaceResult.RedirectAllRequestsTo.Protocol);
      console.log('IndexDocument[Suffix]: %s', result.InterfaceResult.IndexDocument.Suffix);
      console.log('ErrorDocument[Key]: %s', result.InterfaceResult.ErrorDocument.Key);
      console.log('RoutingRules:');
      for (let i = 0; i < result.InterfaceResult.RoutingRules; i++) {
        let routingRule = result.InterfaceResult.RoutingRules[i];
        let condition = routingRule.Condition;
        console.log("Condition[%d]-KeyPrefixEquals:%s, HttpStatusCodeReturnedEquals:%s",
          i, condition.KeyPrefixEquals, condition.HttpStatusCodeReturnedEquals
        );
        let redirect = routingRule.Redirect;
        console.log("Redirect[%d]-Protocol:%s, HostName:%s, ReplaceKeyPrefixWith:%s,
          HttpStatusCode:%s",
```

```
        i, redirect.Protocol, redirect.HostName, redirect.ReplaceKeyPrefixWith, redirect.HttpRedirectCode
    );
    };
    return;
};

    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    } catch (error) {
        console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
        console.log(error);
    };
};
};

getBucketWebsite();
```

15.4 删除托管配置(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

OBS允许在桶内保存静态的网页资源，如.html网页文件、flash文件、音视频文件等，当客户端通过桶的Website接入点访问这些对象资源时，浏览器可以直接解析出这些支持的网页资源，呈现给最终用户。典型的应用场景有：

- 重定向所有的请求到另外一个站点。
- 设定特定的重定向规则来重定向特定的请求。

调用删除桶的网站配置接口，您可以删除指定桶的网站配置。

接口约束

- 您必须是桶所有者或拥有删除桶的网站配置的权限，才能删除桶的网站配置。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:DeleteBucketWebsite权限，如果使用桶策略则需授予DeleteBucketWebsite权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.deleteBucketWebsite(params)
```

请求参数

表 15-23 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释： 桶名。</p> <p>约束限制：</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围： 长度为3~63个字符。</p> <p>默认取值： 无</p>

返回结果

表 15-24 返回结果

参数类型	描述
<p>表15-25</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释： 接口返回信息，详见表15-25。</p>

表 15-25 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表15-27	参数解释: 操作成功后的结果数据，详见 表15-27 。 约束限制: 当Status大于300时为空。

表 15-26 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 15-27 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

您可以通过`ObsClient.deleteBucketWebsite`删除桶的托管配置。以下代码展示了如何删除名为`examplebucket`桶的网站配置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs')
;
// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucketWebsite() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
    };
    // 删除桶的网站配置
    const result = await obsClient.deleteBucketWebsite(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s website configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

deleteBucketWebsite();
```

16 标签管理(Node.js SDK)

16.1 设置桶标签(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

为桶添加标签后，该桶上所有请求产生的计费话单里都会带上这些标签，从而可以针对话单报表做分类筛选，进行更详细的成本分析。例如：某个应用程序在运行过程会往桶里上传数据，我们可以用应用名称作为标签，设置到被使用的桶上。在分析话单时，就可以通过应用名的标签来分析此应用的成本。

调用设置桶标签接口，您可为指定桶添加标签。

更多关于桶标签的内容请参考[桶标签](#)。

接口约束

- 每个桶最多能设置10个标签。
- 标签的键名（Key）的最大长度为36个字符，标签的键值（Value）的最大长度为43个字符。
- 标签的键名（Key）和键值（Value）不能包含字符“,”、“*”、“|”、“/”、“<”、“>”、“=”、“\”以及ASCII码0x00--0x1F这些控制字符。
- 您必须是桶拥有者或拥有设置桶标签的权限，才能设置桶标签。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutBucketTagging权限，如果使用桶策略则需授予PutBucketTagging权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.setBucketTagging(params)
```

请求参数

表 16-1 请求参数列表

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>
Tags	Tag[]	必选	<p>参数解释: 桶标签列表。</p> <p>约束限制:</p> <ul style="list-style-type: none">一个桶默认最大拥有10个标签。每个标签有且只有一对键值。同一个桶标签的键不能重复，但是值可以重复，并且可以为空。 <p>取值范围: 详见Tag。</p> <p>默认取值: 无</p>

表 16-2 Tag

参数名称	参数类型	是否必选	描述
Key	string	作为请求参数时必选	<p>参数解释: 标签的键。</p> <p>约束限制:</p> <ul style="list-style-type: none"> • 同一个桶标签的键不能重复。 • 可以自定义，也可以选择预先在标签管理服务（TMS）创建好的标签。 • 长度范围为1到36个字符。 • 不能以空格开头或结尾，不能包含非打印字ASCII（0-31）、“=”、“*”、“<”、“>”、“\”。 • 区分大小写。 <p>取值范围: 长度为1到36个字符。</p> <p>默认取值: 无</p>
Value	string	作为请求参数时必选	<p>参数解释: 标签的值。</p> <p>约束限制: 标签的值可以重复，并且可以为空。</p> <ul style="list-style-type: none"> • 长度范围为0到43个字符。 • 不能包含非打印字ASCII（0-31）、“=”、“*”、“<”、“>”、“\”。 • 区分大小写。 <p>取值范围: 长度为0到43个字符。</p> <p>默认取值: 无</p>

返回结果

表 16-3 返回结果

参数类型	描述
<p>表16-4</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表16-4。</p>

表 16-4 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表16-6	参数解释: 操作成功后的结果数据，详见 表16-6 。 约束限制: 当Status大于300时为空。

表 16-5 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 16-6 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

以下代码展示了如何设置examplebucket桶的标签配置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK, 这里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, 获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS, 同样建议您尽量避免使用硬编码, 以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint, 这里以中国-香港为例, 其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketTagging() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定桶的标签
      Tags: [
        {Key: "key0", Value: "value0"},
        {Key: "key1", Value: "value1"},
      ]
    };
    // 设置桶标签配置
    const result = await obsClient.setBucketTagging(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s tagging configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

setBucketTagging();
```

16.2 获取桶标签(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

为桶添加标签后，该桶上所有请求产生的计费话单里都会带上这些标签，从而可以针对话单报表做分类筛选，进行更详细的成本分析。例如：某个应用程序在运行过程会往桶里上传数据，我们可以用应用名称作为标签，设置到被使用的桶上。在分析话单时，就可以通过应用名的标签来分析此应用的成本。

调用获取桶标签接口，您可获取指定桶的标签。

更多关于桶标签的内容请参考[桶标签](#)。

接口约束

- 您必须是桶拥有者或拥有获取桶标签的权限，才能获取桶标签。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:GetBucketTagging权限，如果使用桶策略则需授予GetBucketTagging权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.getBucketTagging(params)
```

请求参数

表 16-7 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none"> 桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。 桶命名规则如下： <ul style="list-style-type: none"> 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。 禁止使用IP地址。 禁止以“-”或“.”开头及结尾。 禁止两个“.”相邻（如：“my..bucket”）。 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。 同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

返回结果

表 16-8 返回结果

参数类型	描述
<p>表16-9</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表16-9。</p>

表 16-9 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表16-11	参数解释: 操作成功后的结果数据，详见 表16-11 。 约束限制: 当Status大于300时为空。

表 16-10 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 16-11 GetBucketTaggingOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。
Tags	Tag[]	参数解释: 桶标签列表, 详见Tag。

表 16-12 Tag

参数名称	参数类型	是否必选	描述
Key	string	作为请求参数时必选	参数解释: 标签的键。 约束限制: <ul style="list-style-type: none">• 同一个桶标签的键不能重复。• 可以自定义, 也可以选择预先在标签管理服务 (TMS) 创建好的标签。• 长度范围为1到36个字符。• 不能以空格开头或结尾, 不能包含非打印字ASCII (0-31)、"="、"*"、"<"、">"、"\"。• 区分大小写。 取值范围: 长度为1到36个字符。 默认取值: 无
Value	string	作为请求参数时必选	参数解释: 标签的值。 约束限制: 标签的值可以重复, 并且可以为空。 <ul style="list-style-type: none">• 长度范围为0到43个字符。• 不能包含非打印字ASCII (0-31)、"="、"*"、"<"、">"、"\"。• 区分大小写。 取值范围: 长度为0到43个字符。 默认取值: 无

代码示例

您可以通过ObsClient.getBucketTagging查看桶标签。以下代码展示了如何获取名为examplebucket桶的标签配置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketTagging() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket"
    };
    // 获取桶标签配置
    const result = await obsClient.getBucketTagging(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s tagging configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let tag of result.InterfaceResult.Tags) {
        console.log('Key: %s, Value: %s', tag.key, tag.value);
      }
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};
getBucketTagging();
```

16.3 删除桶标签(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

功能介绍

为桶添加标签后，该桶上所有请求产生的计费话单里都会带上这些标签，从而可以针对话单报表做分类筛选，进行更详细的成本分析。例如：某个应用程序在运行过程会往桶里上传数据，我们可以用应用名称作为标签，设置到被使用的桶上。在分析话单时，就可以通过应用名的标签来分析此应用的成本。

调用删除桶标签接口，您可删除指定桶的标签。

更多关于桶标签的内容请参考[桶标签](#)。

接口约束

- 您必须是桶拥有者或拥有删除桶标签的权限，才能删除桶标签。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:DeleteBucketTagging权限，如果使用桶策略则需授予DeleteBucketTagging权限。相关授权方式介绍可参见[OBS权限控制概述](#)，配置方式详见[使用IAM自定义策略](#)、[自定义创建桶策略](#)。
- OBS支持的Region与Endpoint的对应关系，详细信息请参见[地区与终端节点](#)。

方法定义

```
ObsClient.deleteBucketTagging(params)
```

请求参数

表 16-13 BucketRequestInput

参数名称	参数类型	是否必选	描述
Bucket	string	必选	<p>参数解释: 桶名。</p> <p>约束限制:</p> <ul style="list-style-type: none">桶的名字需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。桶命名规则如下：<ul style="list-style-type: none">3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。禁止使用IP地址。禁止以“-”或“.”开头及结尾。禁止两个“.”相邻（如：“my..bucket”）。禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。同一用户在同一个区域多次创建同名桶不会报错，创建的桶属性以第一次请求为准。 <p>取值范围: 长度为3~63个字符。</p> <p>默认取值: 无</p>

返回结果

表 16-14 返回结果

参数类型	描述
<p>表16-15</p> <p>说明 该接口返回是一个Promise类型，需要使用Promise、async/await语法处理。</p>	<p>参数解释: 接口返回信息，详见表16-15。</p>

表 16-15 Response

参数名称	参数类型	描述
CommonMsg	ICommonMsg	参数解释: 接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等，详见 ICommonMsg 。
InterfaceResult	表16-17	参数解释: 操作成功后的结果数据，详见 表16-17 。 约束限制: 当Status大于300时为空。

表 16-16 ICommonMsg

参数名称	参数类型	描述
Status	number	参数解释: OBS服务端返回的HTTP状态码。 取值范围: 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 。
Code	string	参数解释: OBS服务端返回的错误码。
Message	string	参数解释: OBS服务端返回的错误描述。
HostId	string	参数解释: OBS服务端返回的请求服务端ID。
RequestId	string	参数解释: OBS服务端返回的请求ID。
Id2	string	参数解释: OBS服务端返回的请求ID2。
Indicator	string	参数解释: OBS服务端返回的详细错误码。

表 16-17 BaseResponseOutput

参数名称	参数类型	描述
RequestId	string	参数解释: OBS服务端返回的请求ID。

代码示例

您可以通过ObsClient.deleteBucketTagging删除桶标签。以下代码展示了如何删除名为examplebucket桶的标签配置。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AK/SK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucketTagging() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket"
    };
    // 删除桶标签配置
    const result = await obsClient.deleteBucketTagging(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s tagging configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("Delete bucket(%s)'s tagging configuration fail!", params.Bucket);
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

deleteBucketTagging();
```

17 服务端加密(Node.js SDK)

17.1 服务端加密简介(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

OBS支持服务端加密功能，使对象加密的行为在OBS服务端进行。

更多关于服务端加密的内容请参考[服务端加密](#)。

17.2 加密说明(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

OBS Node.js SDK支持服务端加密的接口见下表：

OBS Node.js SDK接口方法	描述	支持加密类型
ObsClient.putObject	上传对象时设置加密算法、密钥，对对象启用服务端加密。	SSE-KMS SSE-C
ObsClient.appendObject	追加上传对象时设置加密算法、密钥，对对象启用服务端加密。	SSE-KMS SSE-C
ObsClient.getObject	下载对象时设置解密算法、密钥，用于解密对象。	SSE-C

OBS Node.js SDK接口方法	描述	支持加密类型
ObsClient.copyObject	<ol style="list-style-type: none"> 复制对象时设置源对象的解密算法、密钥，用于解密源对象。 复制对象时设置目标对象的加密算法、密钥，对目标对象启用加密算法。 	SSE-KMS SSE-C
ObsClient.getObjectMetadata	获取对象元数据时设置解密算法、密钥，用于解密对象。	SSE-C
ObsClient.initiateMultipartUpload	初始化分段上传任务时设置加密算法、密钥，对分段上传任务最终生成的对象启用服务端加密。	SSE-KMS SSE-C
ObsClient.uploadPart	上传段时设置加密算法、密钥，对分段数据启用服务端加密。	SSE-C
ObsClient.copyPart	<ol style="list-style-type: none"> 复制段时设置源对象的解密算法、密钥，用于解密源对象。 复制段时设置目标段的加密算法、密钥，对目标段启用加密算法。 	SSE-C

OBS Node.js SDK两种加密方式支持的请求参数：

加密类型	OBS Node.js SDK对应请求参数	说明
SSE-KMS	SseKms	表示服务端加密是SSE-KMS方式，目前仅支持： kms 。
	SseKmsKey	表示SSE-KMS方式下的主密钥，可为空。
SSE-C	SseC	表示服务端加密是SSE-C方式，目前仅支持： AES256 。
	SseCKey	表示SSE-C方式下的密钥，由AES256算法得到。上传对象时作为加密密钥；下载对象时作为解密密钥。
	CopySourceSseC	适用于ObsClient.copyObject和ObsClient.copyPart，表示以SSE-C方式解密源对象，目前仅支持： AES256 。
	CopySourceSseCKey	适用于ObsClient.copyObject和ObsClient.copyPart，表示以SSE-C方式解密源对象时使用的密钥，由AES256算法得到。

17.3 加密示例(Node.js SDK)

须知

开发过程中，您有任何问题可以在github上[提交issue](#)。

代码示例一：上传对象加密

本示例使用SSE-C方式加密上传example/objectname到examplebucket桶中。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: "example/objectname",
      // 指定文本对象
      Body: 'Hello OBS',
      // 设置SSE-C算法加密对象
      SseC: 'AES256',
      SseCKey: 'your sse-c key generated by AES-256 algorithm'
    };
    // 加密上传对象
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put object(%s) under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};
```

```
putObject());
```

代码示例二：下载对象解密

本示例用于流式下载examplebucket桶中的example/objectname加密对象。

```
// 引入obs库
// 使用npm安装
const ObsClient = require("esdk-obs-nodejs");
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  // 推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // 【可选】如果使用临时AK/SK和SecurityToken访问OBS，同样建议您尽量避免使用硬编码，以降低信息泄露风险。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
  // security_token: process.env.SECURITY_TOKEN,
  // endpoint填写Bucket对应的Endpoint，这里以中国-香港为例，其他地区请按实际情况填写
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // 指定存储桶名称
      Bucket: "examplebucket",
      // 指定对象名，此处以 example/objectname 为例
      Key: 'example/objectname',
      // 设置SSE-C算法解密对象
      SseC: 'AES256',
      // 此处的密钥必须和上传对象加密时使用的密钥一致
      SseCKey: 'your sse-c key generated by AES-256 algorithm'
    };

    // 下载加密对象
    const result = await obsClient.getObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log('Object Content: %s', result.InterfaceResult.Content);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

getObject();
```


18 异常处理(Node.js SDK)

18.1 OBS 服务端错误码(Node.js SDK)

在向OBS服务端发出请求后，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。详细的错误码及其对应的描述和HTTP状态码见下表：

错误码	描述	HTTP状态码
AccessDenied	拒绝访问。	403 Forbidden
AccessForbidden	权限不足。	403 Forbidden
AccountProblem	用户的账户出现异常（过期、冻结等），不能成功地完成操作。	403 Forbidden
AllAccessDisabled	用户无权限执行某操作。	403 Forbidden
AmbiguousGrantByEmailAddress	用户提供的Email地址关联的账户超过了1个。	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。	400 Bad Request
BadDomainName	域名不合法。	400 Bad Request
BadRequest	请求参数不合法。	400 Bad Request
BucketAlreadyExists	请求的桶名已经存在。桶的命名空间是系统中所有用户共用的，选择一个不同的桶名再重试一次。	409 Conflict
BucketAlreadyOwnedByYou	发起该请求的用户已经创建过了这个名字的桶，并拥有这个桶。	409 Conflict

错误码	描述	HTTP状态码
BucketNotEmpty	用户尝试删除的桶不为空。	409 Conflict
CredentialsNotSupported	该请求不支持证书验证。	400 Bad Request
CustomDomainAlreadyExist	配置了已存在的域。	400 Bad Request
CustomDomainNotExist	操作的域不存在。	400 Bad Request
DeregisterUserId	用户已经注销。	403 Forbidden
EntityTooSmall	用户试图上传的对象大小小于系统允许的最小大小。	400 Bad Request
EntityTooLarge	用户试图上传的对象大小超过了系统允许的最大大小。	400 Bad Request
FrozenUserId	用户被冻结。	403 Forbidden
IllegalVersioningConfigurationException	请求中的版本配置无效。	400 Bad Request
IllegalLocationConstraintException	配置了与所在Region不匹配的区域限制。	400 Bad Request
InArrearOrInsufficientBalance	因为ACL而没有权限进行某种操作。	403 Forbidden
IncompleteBody	请求体不完整。	400 Bad Request
IncorrectNumberOfFilesInPostRequest	每个POST请求都需要带一个上传的文件。	400 Bad Request
InlineDataTooLarge	Inline Data超过了允许的最大长度。	400 Bad Request
InsufficientStorageSpace	存储空间不足。	403 Forbidden
InternalServerError	系统遇到内部错误，请重试。	500 Internal Server Error
InvalidAccessKeyId	系统记录中不存在客户提供的Access Key Id。	403 Forbidden
InvalidAddressingHeader	用户必须指定匿名角色。	N/A
InvalidArgument	无效的参数。	400 Bad Request
InvalidBucketName	请求中指定的桶名无效。	400 Bad Request
InvalidBucket	请求访问的桶已不存在。	400 Bad Request

错误码	描述	HTTP状态码
InvalidBucketState	无效的桶状态。	409 Conflict
InvalidBucketStoragePolicy	修改桶策略时，提供的新策略不合法。	400 Bad Request
InvalidDigest	HTTP头中指定的Content-MD5值无效。	400 Bad Request
InvalidEncryptionAlgorithmError	错误的加密算法。	400 Bad Request
InvalidLocationConstraint	创建桶时，指定的location不合法。	400 Bad Request
InvalidPart	一个或多个指定的段无法找到。这些段可能没有上传，或者指定的entity tag与段的entity tag不一致。	400 Bad Request
InvalidPartOrder	段列表的顺序不是升序，段列表必须按段号升序排列。	400 Bad Request
InvalidPayer	所有对这个对象的访问已经无效了。	403 Forbidden
InvalidPolicyDocument	表单中的内容与策略文档中指定的条件不一致。	400 Bad Request
InvalidRange	请求的range不可获得。	416 Client Requested Range Not Satisfiable
InvalidRedirectLocation	无效的重定向地址。	400 Bad Request
InvalidRequest	无效请求。	400 Bad Request
InvalidRequestBody	POST请求体无效。	400 Bad Request
InvalidSecurity	提供的安全证书无效。	403 Forbidden
InvalidStorageClass	用户指定的Storage Class无效。	400 Bad Request
InvalidTargetBucketForLogging	delivery group对目标桶无ACL权限。	400 Bad Request
InvalidURI	无法解析指定的URI。	400 Bad Request
KeyTooLong	提供的Key过长。	400 Bad Request

错误码	描述	HTTP状态码
MalformedACLError	提供的XML格式错误，或者不符合要求的格式。	400 Bad Request
MalformedError	请求中携带的XML格式不正确。	400 Bad Request
MalformedLoggingStatus	Logging的XML格式不正确。	400 Bad Request
MalformedPolicy	Bucket policy检查不通过。	400 Bad Request
MalformedPOSTRequest	POST请求的请求体不是结构化良好的多段或形式化数据。	400 Bad Request
MalformedQuotaError	Quota的XML格式不正确。	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。错误消息是：“The XML you provided was not well-formed or did not validate against our published schema.”。	400 Bad Request
MaxMessageLengthExceeded	请求消息过长。	400 Bad Request
MaxPostPreDataLengthExceeded Error	在上传文件前面的POST请求域过大。	400 Bad Request
MetadataTooLarge	元数据消息头超过了允许的最大元数据大小。	400 Bad Request
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.	405 Method Not Allowed
MissingContentLength	必须要提供HTTP消息头中的Content-Length字段。	411 Length Required
MissingRegion	请求中缺少Region信息，且系统无默认Region。	400 Bad Request
MissingRequestBodyError	当用户发送一个空的XML文档作为请求时会发生。错误消息是：“Request body is empty.”。	400 Bad Request

错误码	描述	HTTP状态码
MissingRequiredHeader	请求中缺少必要的头域。	400 Bad Request
MissingSecurityHeader	请求缺少一个必须的头。	400 Bad Request
NoSuchBucket	指定的桶不存在。	404 Not Found
NoSuchBucketPolicy	桶policy不存在。	404 Not Found
NoSuchCORSConfiguration	CORS配置不存在。	404 Not Found
NoSuchCustomDomain	请求的用户域不存在。	404 Not Found
NoSuchKey	指定的Key不存在。	404 Not Found
NoSuchLifecycleConfiguration	请求的LifeCycle不存在。	404 Not Found
NoSuchPolicy	给定的policy名字不存在。	404 Not Found
NoSuchUpload	指定的多段上传不存在。 Upload ID不存在，或者多段上传已经终止或完成。	404 Not Found
NoSuchVersion	请求中指定的version ID与现存的所有版本都不匹配。	404 Not Found
NoSuchWebsiteConfiguration	请求的Website不存在。	404 Not Found
NotImplemented	用户提供的消息头功能上还没有实现。	501 Not Implemented
NotSignedUp	账户未在系统中注册，必须先 在系统中注册了才能使用该账户。	403 Forbidden
OperationAborted	另外一个冲突的操作当前正作用在这个资源上，请重试。	409 Conflict
PermanentRedirect	尝试访问的桶必须使用指定的节点，将以后的请求发送到这个节点。	301 Moved Permanently
PreconditionFailed	用户指定的先决条件中至少有一项没有包含。	412 Precondition Failed
Redirect	临时重定向。	307 Moved Temporarily
RequestIsNotMultiPartContent	桶POST必须是闭式的多段/表单数据。	400 Bad Request

错误码	描述	HTTP状态码
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。	400 Bad Request
RequestTimeTooSkewed	请求的时间与服务器的时间相差太大。	403 Forbidden
RequestTorrentOfBucketError	不允许请求桶的torrent文件。	400 Bad Request
ServiceNotImplemented	请求的方法服务端没有实现。	501 Not Implemented
ServiceNotSupported	请求的方法服务端不支持。	409 Conflict
ServiceUnavailable	服务器过载或者内部错误异常。	503 Service Unavailable
SignatureDoesNotMatch	请求中带的签名与系统计算得到的签名不一致。检查您的访问密钥（AK和SK）和签名计算方法。	403 Forbidden
SlowDown	请降低请求频率。	503 Service Unavailable
System Capacity Not enough	系统空间不足异常。	403 Forbidden
TooManyCustomDomains	配置了过多的用户域。	400 Bad Request
TemporaryRedirect	当DNS更新时，请求将被重定向到桶。	307 Moved Temporarily
TooManyBuckets	用户拥有的桶的数量达到了系统的上限，并且请求试图创建一个新桶。	400 Bad Request
TooManyObjectCopied	用户单个对象被复制的数量超过系统上限。	400 Bad Request
TooManyWrongSignature	因高频错误请求被拒绝服务。	400 Bad Request
UnexpectedContent	该请求不支持带内容字段。	400 Bad Request
UnresolvableGrantByEmailAddresses	用户提供的Email与记录中任何账户的都不匹配。	400 Bad Request
UserKeyMustBeSpecified	请求中缺少用户的AK信息。	400 Bad Request
WebsiteRedirect	Website请求缺少bucketName。	301 Moved Permanently

错误码	描述	HTTP状态码
KMS.DisabledException	SSE-KMS加密方式下，主密钥被禁用。	400 Bad Request
KMS.NotFoundException	SSE-KMS加密方式下，主密钥不存在。	400 Bad Request
RestoreAlreadyInProgress	对象正在恢复，请求冲突。	409 Conflict
ObjectHasAlreadyRestored	已经恢复的对象，禁止缩短恢复保存时间。	409 Conflict
InvalidObjectState	恢复对象不是归档存储对象。	403 Forbidden
InvalidTagError	配置桶标签时，提供了无效的Tag。	400 Bad Request
NoSuchTagSet	指定的桶没有设置标签。	404 Not Found

18.2 SDK 公共结果对象(Node.js SDK)

调用ObsClient的相关接口完成后，如果未发生异常，则均会返回公共结果对象。该对象包含的内容见下表：

参数名称	参数类型	描述
CommonMsg	Object	接口调用完成后的公共信息，包含HTTP状态码，操作失败的错误码等。
-	Status	Number HTTP状态码，小于300表明操作成功；反之，表明操作失败。
-	Code	String OBS服务端错误码，当Status小于300时为空。
-	Message	String OBS服务端错误描述，当Status小于300时为空。
-	HostId	String 请求的服务端ID，当Status小于300时为空。
-	RequestId	String OBS服务端返回的请求ID。
-	Id2	String OBS服务端返回的请求ID2。
-	Indicator	String OBS服务端返回的详细错误码，当Status小于300时为空。
InterfaceResult	Object	操作成功后的结果数据，当Status大于300时为空。
-	RequestId	String OBS服务端返回的请求ID。
-	Id2	String OBS服务端返回的请求ID2。

参数名称	参数类型	描述
其他字段		请查阅对应章节的返回结果列表。

处理公共结果对象的代码示例如下：

```
// 引入obs库
// 使用npm安装
var ObsClient = require('esdk-obs-nodejs');
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  //推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca\_01\_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  //这里以中国-香港为例，其他地区请按实际情况填写
  server: 'https://obs.ap-southeast-1.myhuaweicloud.com'
});

// 调用接口进行操作，例如下载对象
obsClient.getObject({
  Bucket: 'bucketname',
  Key: 'objectname',
}, (err, result) => {
  if(!err){
    if(result.CommonMsg.Status < 300){
      // 获取RequestId
      console.log('RequestId-->' + result.InterfaceResult.RequestId);
      // 获取其他参数
      console.log('Content-->' + result.InterfaceResult.Content.toString());
    }else{
      // 获取Code和Message
      console.log('Code-->' + result.CommonMsg.Code);
      console.log('Message-->' + result.CommonMsg.Message);
    }
  }
});
```

18.3 日志分析(Node.js SDK)

日志配置

OBS Node.js SDK基于log4js开源库提供了日志功能，您可以通过ObsClient.initLog开启日志功能并进行配置。代码示例如下：

```
obsClient.initLog({
  name: 'test', // 日志名称
  file_full_path: './logs/OBS-SDK.log', // 配置日志文件路径
  max_log_size: 20480, // 配置日志文件大小，单位：字节
  backups: 10, // 配置最大可保留的日志文件个数
  level: 'warn', // 配置日志级别
  log_to_console: true // 配置是否将日志打印到console
});
```


📖 说明

- 日志功能默认是关闭的，需要主动开启。
- 使用`file_full_path`参数配置日志文件路径，可配置相对路径或绝对路径。

日志内容格式

SDK日志格式为：日志时间|日志级别|调用接口|日志内容。示例如下：

```
2017/10/12 10:21:05 666|INFO |ListBuckets|enter ListBuckets...
2017/10/12 10:21:05 672|INFO |ListBuckets|prepare request parameters ok,then Send request to service start
2017/10/12 10:21:05 715|INFO |ListBuckets|2017-10-12 10:21:05|http cost 34 ms|0|
2017/10/12 10:21:05 716|INFO |ListBuckets|get response start, statusCode:200
```

日志级别

当系统出现问题需要定位且当前的日志无法满足要求时，可以通过修改日志的级别来获取更多的信息。其中debug日志信息最丰富，error日志信息最少。

具体说明如下：

- debug：调试级别，如果设置为这个级别，将打印SDK记录的所有日志。
- info：信息级别，如果设置为这个级别，除了打印warn级别的信息外，还将打印HTTP/HTTPS请求的耗时时间等信息。
- warn：告警级别，如果设置为这个级别，除了打印error级别的信息外，还将打印一些关键事件的信息。
- error：错误级别，如果设置为这个级别，仅打印发生异常时的错误信息。

18.4 缺少模块异常(Node.js SDK)

异常说明：使用OBS Node.js SDK进行二次开发时如果报缺少模块异常，如“Cannot find module 'xml2js'”。

解决方案：

1. 请检查本地Node.js SDK版本号，是否为最新版本。
2. 请确保依赖库已正确安装，参见[安装SDK](#)。

18.5 连接超时异常(Node.js SDK)

异常说明：调用接口时出现“connect ETIMEDOUT”错误，表明连接超时，其原因一般是服务地址（Endpoint）错误或网络不通导致无法连接OBS服务。

解决方案：

1. 请检查服务地址是否为空，或者是非法的服务器地址。
2. 请检查网络状况。
3. 检查服务端是否返回了异常的结果，并联系服务端运维团队。

18.6 签名不匹配异常(Node.js SDK)

异常说明： 如果从返回结果的CommonMsg.Status中获取到的HTTP状态码为403，CommonMsg.Code中获取到的OBS服务端错误码为SignatureDoesNotMatch。

解决方案：

1. 检查endpoint：是否在endpoint前多加了桶名，这个也会报签名不匹配错误。
2. 检查AK、SK：请确保AK、SK输入正确，成对匹配，且与请求所用AK、SK保持一致。
3. 检查StringToSign：

StringToSign的构造规则：

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

19 常见问题(Node.js SDK)

19.1 如何指定 Content-SHA256?(Node.js SDK)

上传对象和上传段支持携带x-obs-content-sha256头域，该头域值为请求消息体256-bit SHA256值转十六进制值，计算方式为Hex(SHA256Hash(<payload>)，服务端会对携带此头域的请求计算其消息体的sha256值做校验（性能会有部分下降，在安全上推荐该算法），上传对象代码示例如下：

```
// 引入obs库
// 使用npm安装
var ObsClient = require('esdk-obs-nodejs');
var crypto = require('crypto');
var fs = require('fs');
// 使用源码安装
// var ObsClient = require('./lib/obs');

// 创建ObsClient实例
const obsClient = new ObsClient({
  //推荐通过环境变量获取AKSK，这里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，获取方式请参见https://support.huaweicloud.com/intl/zh-cn/usermanual-ca/ca_01_0003.html
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  //这里以中国-香港为例，其他地区请按实际情况填写
  server: 'https://obs.ap-southeast-1.myhuaweicloud.com'
});

let filePath = 'D:\\example.xml'
let sha256 = crypto.createHash('sha256').update(fs.readFileSync(filePath, 'utf8'), 'utf8').digest('hex')

obsClient.putObject({
  Bucket: 'bucketname',
  Key: 'objectname',
  SourceFile: filePath, // filePath为待上传的本地文件路径，需要指定到具体的文件名
  ContentSha256: sha256
}, (err, result) => {
  if(err){
    console.error('Error-->' + err);
  }else{
    console.log('Status-->' + result.CommonMsg.Status);
  }
});
```

须知

Node.js SDK 同时支持MD5与SHA256校验，在安全上更推荐使用SHA256算法

19.2 为什么 SDK 源码中包含 acs.amazonaws.com 关键字? (Node.js SDK)

OBS SDK 为了兼容访问aws s3服务，在源码中会包含acs.amazonaws.com关键字的固定格式，SDK仅作常量使用，不会向该格式有其他处理，也不会对其访问。SDK会自动兼容该场景，用户可不感知。比如[Node.js SDK](#):

```
Utils.prototype.URIAdapter = function(value, signatureContext){
    value = value || '';
    value = String(value);
    if(signatureContext.signature === 'obs'){
        if(obsAllowedUri.indexOf(value) >= 0){
            return value;
        }
        if(value === 'AllUsers' || value === 'http://acs.amazonaws.com/groups/global/AllUsers'){
            return 'Everyone';
        }
    }
    return '';
}

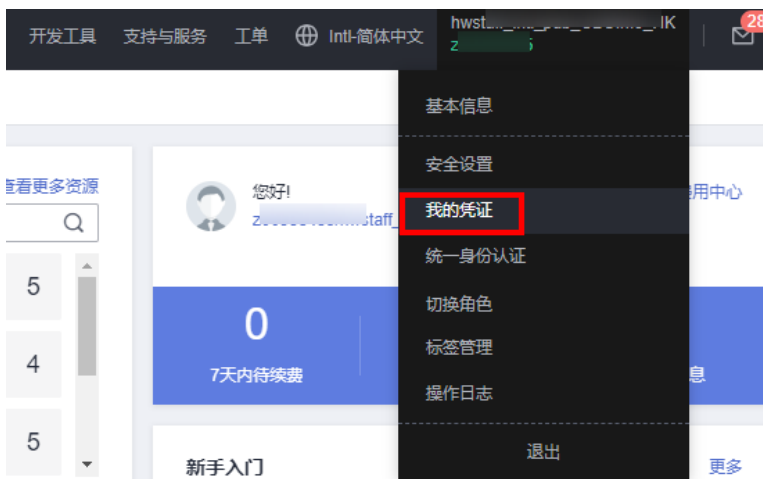
if(v2AllowedUri.indexOf(value) >= 0){
    return value;
}
if(value === 'Everyone' || value === 'AllUsers'){
    return 'http://acs.amazonaws.com/groups/global/AllUsers';
}
if(value === 'AuthenticatedUsers'){
    return 'http://acs.amazonaws.com/groups/global/AuthenticatedUsers';
}
if(value === 'LogDelivery'){
    return 'http://acs.amazonaws.com/groups/s3/LogDelivery';
}
return '';
```

19.3 如何获取账号 ID 和用户 ID? (Node.js SDK)

获取账号、IAM 用户、项目的名称和 ID

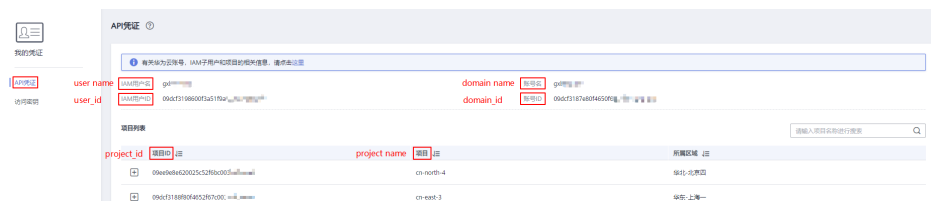
- 从控制台获取账号名、账号ID、用户名、用户ID、项目名称、项目ID
 - a. 在华为云首页右上角，单击“控制台”。
 - b. 在右上角的用户名中选择“我的凭证”。

图 19-1 进入我的凭证



- c. 在“我的凭证”界面，API凭证页签中，查看账号名、账号ID、用户名、用户ID、项目名称、项目ID。
每个区域的项目ID有所不同，需要根据业务所在的区域获取对应的项目ID。

图 19-2 查看账号名、账号 ID、用户名、用户 ID、项目名称、项目 ID



- 调用API获取用户ID、项目ID
 - 获取用户ID请参考：[管理员查询IAM用户列表](#)。
 - 获取项目ID请参考：[查询指定条件下的项目列表](#)。

获取用户组名称和 ID

步骤1 登录华为云，进入IAM控制台，选择“用户组”页签。

步骤2 单击需要查询的用户组前的下拉框，即可查询用户组名称、用户组ID。

----结束

获取区域 ID

步骤1 登录华为云，进入IAM控制台，选择“项目”页签。

步骤2 “项目”列的内容即为所属区域对应的ID。

----结束

获取委托名称和 ID

步骤1 登录华为云，进入IAM控制台，选择“委托”页签。

步骤2 鼠标移动到需要查询名称和ID的委托上，黑色框中出现的第一行为委托名称，第二行为委托ID。

---结束

19.4 如何理解 Content-Type (MIME) (Node.js SDK)

Content-Type (MIME) 用于标识发送或接收数据的类型，浏览器根据该参数来决定数据的打开方式。Content-Type多用于指定一些客户端自定义的文件，以及一些媒体文件的打开方式。

📖 说明

如果您在上传Object时未指定Content-Type，SDK会根据指定Object名称的后缀名来判定文件类型并自动填充Content-Type。

什么是 Content-Type

Content-Type用于指定Object的文件类型。内容类型决定浏览器将以什么形式、什么编码读取文件。如果没有指定文件类型，则根据文件的扩展名生成。如果文件没有扩展名，则文件类型的默认值application/octet-stream。

常见的 Content-Type

常见的Content-Type (MIME) 列表如下：

表 19-1 常见的 Content-Type (MIME) 列表

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.* (二进制流, 未知的文件类型)	application/octet-stream	.tif	image/tiff
.a11	application/x-a11	.acp	audio/x-mei-aac
.ai	application/postscript	.aif	audio/aiff
.aifc	audio/aiff	.aiff	audio/aiff
.anv	application/x-anv	.apk	application/vnd.android.package-archive
.asa	text/asa	.asf	video/x-ms-asf
.asp	text/asp	.asx	video/x-ms-asf
.au	audio/basic	.avi	video/avi
.awf	application/vnd.adobe.workflow	.biz	text/xml

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.bmp	application/x-bmp	.bot	application/x-bot
.c4t	application/x-c4t	.c90	application/x-c90
.cal	application/x-cals	.cat	application/vnd.ms-pki.seccat
.cdf	application/x-netcdf	.cdr	application/x-cdr
.cel	application/x-cel	.cer	application/x-x509-ca-cert
.cg4	application/x-g4	.cgm	application/x-cgm
.cit	application/x-cit	.class	java/
.cml	text/xml	.cmp	application/x-cmp
.cmx	application/x-cmx	.cot	application/x-cot
.crl	application/pkix-crl	.crt	application/x-x509-ca-cert
.csi	application/x-csi	.css	text/css
.csv	text/csv	.cut	application/x-cut
.dbf	application/x-dbf	.dbm	application/x-dbm
.dbx	application/x-dbx	.dcd	text/xml
.dcx	application/x-dcx	.der	application/x-x509-ca-cert
.dgn	application/x-dgn	.dib	application/x-dib
.dll	application/x-msdownload	.doc	application/msword
.docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document	.dot	application/msword
.dotx	application/vnd.openxmlformats-officedocument.wordprocessingml.template	.drw	application/x-drw

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.dtd	text/xml	.dwf	Model/vnd.dwf
.dwf	application/x-dwf	.dwg	application/x-dwg
.dxb	application/x-dxb	.dxf	application/x-dxf
.edn	application/vnd.adobe.edn	.emf	application/x-emf
.eml	message/rfc822	.ent	text/xml
.epi	application/x-epi	.eps	application/x-ps
.eps	application/postscript	.etd	application/x-ebx
.exe	application/x-msdownload	.fax	image/fax
.fdf	application/vnd.fdf	.fif	application/fractals
.fo	text/xml	.frm	application/x-frm
.g4	application/x-g4	.gbr	application/x-gbr
.	application/x-	.gif	image/gif
.gl2	application/x-gl2	.gp4	application/x-gp4
.hgl	application/x-hgl	.hmr	application/x-hmr
.hpg	application/x-hpgl	.hpl	application/x-hpl
.hqx	application/mac-binhex40	.hrf	application/x-hrf
.hta	application/hta	.htc	text/x-component
.htm	text/html	.html	text/html
.htt	text/webviewhtml	.htx	text/html
.icb	application/x-icb	.ico	image/x-icon
.ico	application/x-ico	.iff	application/x-iff
.ig4	application/x-g4	.igs	application/x-igs
.iii	application/x-iphone	.img	application/x-img
.ins	application/x-internet-signup	.ipa	application/vnd.iphone

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.isp	application/x-internet-signup	.IVF	video/x-ivf
.java	java/*	.jfif	image/jpeg
.jpe	image/jpeg	.jpe	application/x-jpe
.jpeg	image/jpeg	.jpg	image/jpeg
.jpg	application/x-jpg	.js	application/x-javascript
.jsp	text/html	.la1	audio/x-liquid-file
.lar	application/x-laplayer-reg	.latex	application/x-latex
.lavs	audio/x-liquid-secure	.lbm	application/x-lbm
.lmsff	audio/x-la-lms	.ls	application/x-javascript
.ltr	application/x-ltr	.m1v	video/x-mpeg
.m2v	video/x-mpeg	.m3u	audio/mpegurl
.m4e	video/mpeg4	.mac	application/x-mac
.man	application/x-troff-man	.math	text/xml
.mdb	application/msaccess	.mdb	application/x-mdb
.mfp	application/x-shockwave-flash	.mht	message/rfc822
.mhtml	message/rfc822	.mi	application/x-mi
.mid	audio/mid	.midi	audio/mid
.mil	application/x-mil	.mml	text/xml
.mnd	audio/x-musicnet-download	.mns	audio/x-musicnet-stream
.mocha	application/x-javascript	.mov	video/quicktime
.movie	video/x-sgi-movie	mp1	audio/mp1
.mp2	audio/mp2	.mp2v	video/mpeg
.mp3	audio/mp3	.mp4	video/mp4

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.mpa	video/x-mpg	.mpd	application/vnd.ms-project
.mpe	video/x-mpeg	.mpeg	video/mpg
.mpg	video/mpg	.mpga	audio/rn-mpeg
.mpp	application/vnd.ms-project	.mps	video/x-mpeg
.mpt	application/vnd.ms-project	.mpv	video/mpg
.mpv2	video/mpeg	.mpw	application/vnd.ms-project
.mpx	application/vnd.ms-project	.mtx	text/xml
.mxx	application/x-mmxp	.net	image/pnetvue
.nrf	application/x-nrf	.nws	message/rfc822
.odc	text/x-ms-odc	.out	application/x-out
.p10	application/pkcs10	.p12	application/x-pkcs12
.p7b	application/x-pkcs7-certificates	.p7c	application/pkcs7-mime
.p7m	application/pkcs7-mime	.p7r	application/x-pkcs7-certreqresp
.p7s	application/pkcs7-signature	.pc5	application/x-pc5
.pci	application/x-pci	.pcl	application/x-pcl
.pcx	application/x-pcx	.pdf	application/pdf
.pdb	chemical/x-pdb	.pdx	application/vnd.adobe.pdx
.pfx	application/x-pkcs12	.pgl	application/x-pgl
.pic	application/x-pic	.pko	application/vnd.ms-pki.pko
.pl	application/x-perl	.plg	text/html
.pls	audio/scpls	.plt	application/x-plt

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.png	image/png	.png	application/x-png
.pot	application/vnd.ms-powerpoint	.potx	application/vnd.openxmlformats-officedocument.presentationml.template
.ppa	application/vnd.ms-powerpoint	.ppm	application/x-ppm
.pps	application/vnd.ms-powerpoint	.ppsx	application/vnd.openxmlformats-officedocument.presentationml.slide show
.ppt	application/vnd.ms-powerpoint	.ppt	application/x-ppt
.pptx	application/vnd.openxmlformats-officedocument.presentationml.presentation	.pr	application/x-pr
.prf	application/pics-rules	.prn	application/x-prn
.prt	application/x-prt	.ps	application/x-ps
.ps	application/postscript	.ptn	application/x-ptn
.pwz	application/vnd.ms-powerpoint	.r3t	text/vnd.rn-realtex3d
.ra	audio/vnd.rn-realaudio	.ram	audio/x-pn-realaudio
.ras	application/x-ras	.rat	application/rat-file
.rdf	text/xml	.rec	application/vnd.rn-recording

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.red	application/x-red	.rgb	application/x-rgb
.rjs	application/vnd.rn-realsystem-rjs	.rjt	application/vnd.rn-realsystem-rjt
.rlc	application/x-rlc	.rle	application/x-rle
.rm	application/vnd.rn-realmedia	.rmf	application/vnd.adobe.rmf
.rmi	audio/mid	.rmj	application/vnd.rn-realsystem-rmj
.rmm	audio/x-pn-realaudio	.rmp	application/vnd.rn-rn_music_package
.rms	application/vnd.rn-realmedia-secure	.rmvb	application/vnd.rn-realmedia-vbr
.rmx	application/vnd.rn-realsystem-rmx	.rnx	application/vnd.rn-realplayer
.rp	image/vnd.rn-realpix	.rpm	audio/x-pn-realaudio-plugin
.rsml	application/vnd.rn-rsml	.rt	text/vnd.rn-realtxt
.rtf	application/msword	.rtf	application/x-rtf
.rv	video/vnd.rn-realvideo	.sam	application/x-sam
.sat	application/x-sat	.sdp	application/sdp
.sdw	application/x-sdw	.sis	application/vnd.symbian.install
.sisx	application/vnd.symbian.install	.sit	application/x-stuffit
.slb	application/x-slb	.sld	application/x-sld

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.sldx	application/vnd.openxmlformats-officedocument.presentationml.slide	.slk	drawing/x-slk
.smi	application/smil	.smil	application/smil
.smk	application/x-smk	.snd	audio/basic
.sol	text/plain	.sor	text/plain
.spc	application/x-pkcs7-certificates	.spl	application/futuresplash
.spp	text/xml	.ssm	application/streamingmedia
.sst	application/vnd.ms-pki.certstore	.stl	application/vnd.ms-pki.stl
.stm	text/html	.sty	application/x-sty
.svg	image/svg+xml	.swf	application/x-shockwave-flash
.tdf	application/x-tdf	.tg4	application/x-tg4
.tga	application/x-tga	.tif	image/tiff
.tif	application/x-tif	.tiff	image/tiff
.tld	text/xml	.top	drawing/x-top
.torrent	application/x-bittorrent	.tsd	text/xml
.txt	text/plain	.uin	application/x-icq
.uls	text/iuls	.vcf	text/x-vcard
.vda	application/x-vda	.vdx	application/vnd.visio
.vml	text/xml	.vpg	application/x-vpeg005
.vsd	application/vnd.visio	.vsd	application/x-vsdx
.vss	application/vnd.visio	.vst	application/vnd.visio

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.vst	application/x-vst	.vsw	application/vnd.visio
.vsx	application/vnd.visio	.vtx	application/vnd.visio
.vxml	text/xml	.wav	audio/wav
.wax	audio/x-ms-wax	.wb1	application/x-wb1
.wb2	application/x-wb2	.wb3	application/x-wb3
.wbmp	image/vnd.wap.wbmp	.wiz	application/msword
.wk3	application/x-wk3	.wk4	application/x-wk4
.wkq	application/x-wkq	.wks	application/x-wks
.wm	video/x-ms-wm	.wma	audio/x-ms-wma
.wmd	application/x-ms-wmd	.wmf	application/x-wmf
.wml	text/vnd.wap.wml	.wmv	video/x-ms-wmv
.wmx	video/x-ms-wmx	.wmz	application/x-ms-wmz
.wp6	application/x-wp6	.wpd	application/x-wpd
.wpg	application/x-wpg	.wpl	application/vnd.ms-wpl
.wq1	application/x-wq1	.wr1	application/x-wr1
.wri	application/x-wri	.wrk	application/x-wrk
.ws	application/x-ws	.ws2	application/x-ws
.wsc	text/scriptlet	.wsdl	text/xml
.wvx	video/x-ms-wvx	.xap	application/x-silverlight-app
.x_b	application/x-x_b	.xdp	application/vnd.adobe.xdp
.xdr	text/xml	.xfd	application/vnd.adobe.xfd
.xdf	application/vnd.adobe.xfdf	.xhtml	text/html

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.xls	application/vnd.ms-excel	.xls	application/x-xls
.xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	.xltx	application/vnd.openxmlformats-officedocument.spreadsheetml.template
.xlw	application/x-xlw	.xml	text/xml
.xpl	audio/scpls	.xq	text/xml
.xql	text/xml	.xquery	text/xml
.xsd	text/xml	.xsl	text/xml
.xslt	text/xml	.xwd	application/x-xwd
.x_t	application/x-x_t	.yaml	text/vnd.yaml
.yml	text/vnd.yml	.webp	image/webp
.tar	application/x-tar	.zip	application/zip