

对象存储服务

# Java SDK 开发指南

文档版本 33  
发布日期 2024-04-18



版权所有 © 华为技术有限公司 2024。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

<b>1 使用前须知(Java SDK)</b> .....	<b>1</b>
<b>2 Java SDK 接口概览(Java SDK)</b> .....	<b>4</b>
<b>3 使用前准备(Java SDK)</b> .....	<b>16</b>
<b>4 下载与安装 SDK(Java SDK)</b> .....	<b>18</b>
<b>5 快速入门(Java SDK)</b> .....	<b>22</b>
<b>6 初始化(Java SDK)</b> .....	<b>28</b>
6.1 初始化概述(Java SDK).....	28
6.2 创建并配置客户端(Java SDK).....	29
6.3 配置 SDK 日志(Java SDK).....	39
<b>7 管理桶(Java SDK)</b> .....	<b>41</b>
7.1 创建桶(Java SDK).....	41
7.2 列举桶(Java SDK).....	56
7.3 删除桶(Java SDK).....	67
7.4 判断桶是否存在(Java SDK).....	70
7.5 获取桶元数据(Java SDK).....	73
7.6 设置桶 ACL(Java SDK).....	80
7.7 获取桶 ACL(Java SDK).....	90
7.8 设置桶策略(Java SDK).....	96
7.9 获取桶策略(Java SDK).....	100
7.10 删除桶策略(Java SDK).....	103
7.11 获取桶区域位置(Java SDK).....	106
7.12 获取桶存量信息(Java SDK).....	109
7.13 设置桶配额(Java SDK).....	112
7.14 获取桶配额(Java SDK).....	116
7.15 设置桶存储类别(Java SDK).....	119
7.16 获取桶存储类别(Java SDK).....	123
7.17 桶清单(Java SDK).....	126
7.17.1 设置桶清单规则(Java SDK).....	126
7.17.2 获取桶清单规则(Java SDK).....	132
7.17.3 列举桶清单规则(Java SDK).....	138
7.17.4 删除桶清单规则(Java SDK).....	144

<b>8 管理并行文件系统(Java SDK)</b> .....	<b>148</b>
8.1 创建并行文件系统(Java SDK).....	148
8.2 列举并行文件系统(Java SDK).....	162
8.3 列举并行文件系统内对象(Java SDK).....	173
8.4 修改写对象(Java SDK).....	188
8.5 重命名对象(Java SDK).....	192
8.6 截断对象(Java SDK).....	196
8.7 设置对象 AccessLabel.....	200
8.8 获取对象 AccessLabel.....	203
8.9 删除对象 AccessLabel.....	207
<b>9 上传对象(Java SDK)</b> .....	<b>211</b>
9.1 对象上传简介(Java SDK).....	211
9.2 流式上传(Java SDK).....	211
9.3 文件上传(Java SDK).....	234
9.4 获取上传进度(Java SDK).....	253
9.5 创建文件夹(Java SDK).....	256
9.6 设置对象元数据(Java SDK).....	277
9.7 分段上传(Java SDK).....	301
9.7.1 分段上传概述(Java SDK).....	301
9.7.2 初始化分段上传任务(Java SDK).....	301
9.7.3 上传段(Java SDK).....	320
9.7.4 合并段(Java SDK).....	330
9.7.5 分段上传场景化示例(Java SDK).....	337
9.7.6 取消分段上传任务(Java SDK).....	339
9.7.7 列举已上传的段(Java SDK).....	343
9.7.8 列举分段上传任务(Java SDK).....	352
9.8 设置对象生命周期(Java SDK).....	365
9.9 追加上传(Java SDK).....	386
9.10 断点续传上传(Java SDK).....	407
9.11 基于表单上传(Java SDK).....	428
<b>10 下载对象(Java SDK)</b> .....	<b>434</b>
10.1 对象下载简介(Java SDK).....	434
10.2 流式下载(Java SDK).....	435
10.3 范围下载(Java SDK).....	450
10.4 获取下载进度(Java SDK).....	464
10.5 限定条件下载(Java SDK).....	467
10.6 重写响应头(Java SDK).....	482
10.7 获取自定义元数据(Java SDK).....	497
10.8 恢复归档存储对象(Java SDK).....	511
10.9 断点续传下载(Java SDK).....	516
10.10 下载对象接口实现图片处理(Java SDK).....	526
10.11 临时授权方式实现图片处理(Java SDK).....	541

<b>11 管理对象(Java SDK)</b> .....	<b>549</b>
11.1 设置对象元数据(Java SDK).....	549
11.2 获取对象元数据(Java SDK).....	561
11.3 设置对象 ACL(Java SDK).....	570
11.4 获取对象 ACL(Java SDK).....	581
11.5 列举对象(Java SDK).....	588
11.6 删除对象(Java SDK).....	608
11.7 批量删除对象(Java SDK).....	613
11.8 复制对象(Java SDK).....	620
11.9 分段复制(Java SDK).....	644
11.10 判断对象是否存在(Java SDK).....	653
<b>12 临时授权访问(Java SDK)</b> .....	<b>659</b>
12.1 通过临时 URL 访问 OBS(Java SDK).....	659
12.2 通过 Header 鉴权访问 OBS(Java SDK).....	680
<b>13 多版本控制(Java SDK)</b> .....	<b>687</b>
13.1 多版本控制简介(Java SDK).....	687
13.2 设置桶多版本状态(Java SDK).....	687
13.3 获取桶多版本状态(Java SDK).....	691
13.4 获取多版本对象(Java SDK).....	694
13.5 复制多版本对象(Java SDK).....	709
13.6 恢复多版本归档存储对象(Java SDK).....	727
13.7 列举多版本对象(Java SDK).....	732
13.8 设置多版本对象 ACL(Java SDK).....	756
13.9 获取多版本对象权限(Java SDK).....	765
13.10 删除多版本对象(Java SDK).....	771
13.11 批量删除多版本对象(Java SDK).....	775
<b>14 生命周期管理(Java SDK)</b> .....	<b>783</b>
14.1 生命周期管理简介(Java SDK).....	783
14.2 设置生命周期规则(Java SDK).....	784
14.3 获取生命周期规则(Java SDK).....	794
14.4 删除生命周期规则(Java SDK).....	803
<b>15 跨域资源共享(Java SDK)</b> .....	<b>807</b>
15.1 跨域资源共享简介(Java SDK).....	807
15.2 设置跨域资源共享规则(Java SDK).....	807
15.3 获取跨域资源共享规则(Java SDK).....	813
15.4 删除跨域资源共享规则(Java SDK).....	818
<b>16 设置访问日志(Java SDK)</b> .....	<b>823</b>
16.1 日志简介(Java SDK).....	823
16.2 设置桶日志规则(Java SDK).....	823
16.3 获取桶日志规则(Java SDK).....	833

<b>17 静态网站托管(Java SDK)</b> .....	<b>841</b>
17.1 静态网站托管简介(Java SDK).....	841
17.2 网站文件托管(Java SDK).....	841
17.3 设置托管配置(Java SDK).....	864
17.4 获取托管配置(Java SDK).....	874
17.5 删除托管配置(Java SDK).....	881
<b>18 桶标签管理(Java SDK)</b> .....	<b>885</b>
18.1 桶标签简介(Java SDK).....	885
18.2 设置桶标签(Java SDK).....	885
18.3 获取桶标签(Java SDK).....	890
18.4 删除桶标签(Java SDK).....	894
<b>19 服务端加密(Java SDK)</b> .....	<b>899</b>
19.1 服务端加密简介(Java SDK).....	899
19.2 加密说明(Java SDK).....	899
19.3 加密示例(Java SDK).....	900
<b>20 客户端加密(Java SDK)</b> .....	<b>903</b>
20.1 客户端加密简介(Java SDK).....	903
20.2 客户端加密接口(Java SDK).....	905
<b>21 问题定位(Java SDK)</b> .....	<b>913</b>
21.1 问题定位方法(Java SDK).....	913
21.2 依赖缺失和依赖冲突的解决(Java SDK).....	914
21.3 其他高频问题汇总(Java SDK).....	915
<b>22 异常处理(Java SDK)</b> .....	<b>919</b>
22.1 HTTP 状态码(Java SDK).....	919
22.2 OBS 服务端错误码(Java SDK).....	920
22.3 SDK 自定义异常(Java SDK).....	927
22.4 SDK 公共响应头(Java SDK).....	927
22.5 日志分析(Java SDK).....	928
<b>23 常见问题(Java SDK)</b> .....	<b>930</b>
23.1 如何使对象可以被匿名用户访问? (Java SDK).....	930
23.2 SDK 的重试机制是什么? (Java SDK).....	930
23.3 如何获取桶的静态网站访问地址? (Java SDK).....	930
23.4 如何获取对象 URL? (Java SDK).....	931
23.5 公网环境下如何提高上传大文件速度? (Java SDK).....	931
23.6 如何停止正在上传中的任务? (Java SDK).....	931
23.7 如何进行分段上传? (Java SDK).....	932
23.8 如何进行分段下载? (Java SDK).....	932
23.9 如果桶内已存在同名对象, 如何判定覆盖上传成功? (Java SDK).....	932
23.10 对于加密类型的对象如何使用 URL 进行下载? (Java SDK).....	933
23.11 如何生成 SSE-C 方式的加密密钥(Java SDK).....	933

---

23.12 如何获取 SecurityToken? (Java SDK).....	934
23.13 SDK 是否支持批量上传、下载或复制对象? (Java SDK).....	934
23.14 如何指定 Content-SHA256?(Java SDK).....	935
23.15 为什么 SDK 源码中包含 acs.amazonaws.com 关键字? (Java SDK).....	937
23.16 如何理解 Content-Type ( MIME ) ? (Java SDK).....	937
23.17 如何获取账号 ID 和用户 ID?.....	946
<b>A 修订记录(Java SDK).....</b>	<b>949</b>



# 1 使用前须知(Java SDK)

本文介绍Java SDK的版本变更，并提供版本兼容性说明，以及其他使用前须知。

## 变更说明

如表1所示，本节将为您展示Java SDK的版本变更情况和兼容性说明。

表 1-1 Java SDK 版本变更及兼容性说明

版本	变更类型	说明	是否兼容
<a href="#">v3.23.9.1</a>	新功能 适配第三方组件	新功能： • 支持设置自定义dns解析器 适配第三方组件： • 移除 java-xmlbuilder，使用默认的 javax.xml库	-
<a href="#">v3.23.9</a>	新功能 适配第三方组件	新功能： • 新增配置桶清单接口 • 新增客户端加密 • 支持在生命周期规则中配置碎片过期时间 适配第三方组件： • 使用 okio 3.5.0 替代 okio 2.10.0 • 使用 okhttp 4.11.0 替代 okhttp 4.10.0	是
<a href="#">v3.23.5</a>	新功能	新功能： • 新增双写桶功能 • 支持标准存储、低频存储、归档存储 • 三种桶容量的统计	是

版本	变更类型	说明	是否兼容
<a href="#">v3.23.3</a>	新功能 适配第三方组件	新功能： <ul style="list-style-type: none"><li>支持crr进度查询</li><li>新增对象标签接口（设置、获取、删除对象标签）</li></ul> 适配第三方组件： <ul style="list-style-type: none"><li>使用 powermock-module-junit4 2.0.9 替代 powermock-module-junit4 1.6.5</li><li>使用 powermock-api-mockito2 2.0.9 替代 powermock-api-mockito 1.6.5</li><li>使用 mockito-core 4.11.0 替代 mockito-core 1.10.19</li></ul>	是
<a href="#">v3.22.12</a>	新功能 适配第三方组件	新功能： <ul style="list-style-type: none"><li>Java SDK 实现 posix accessible相关接口</li></ul> 适配第三方组件： <ul style="list-style-type: none"><li>使用 log4j2 2.18.0 替代 log4j2 2.17.1</li><li>使用 okhttp 4.10.0 替代 okhttp 4.9.3</li><li>使用 jackson-core 2.13.3 替代 jackson-core 2.13.0</li><li>使用 jackson-databind 2.13.4.1 替代 jackson-databind 2.13.0</li><li>使用 jackson-annotations 2.13.3 替代 jackson-annotations 2.13.0</li></ul>	是
<a href="#">v3.22.3</a>	适配第三方组件	<ul style="list-style-type: none"><li>使用 log4j2 2.17.1 替代 log4j2 2.17.0</li><li>使用 okhttp 4.9.3 替代 okhttp 4.9.1</li><li>使用 okio 2.10.0 替代 okio 2.7.0</li><li>使用 jackson-core 2.13.0 替代 jackson-core 2.12.5</li><li>使用 jackson-databind 2.13.0 替代 jackson-databind 2.12.5</li><li>使用 jackson-annotations 2.13.0 替代 jackson-annotations 2.12.5</li></ul>	是
<a href="#">v3.21.12</a>	适配第三方组件	使用 log4j2 2.17.0 替代 log4j2 2.16.0	是

版本	变更类型	说明	是否兼容
v3.21.11	新功能 适配第三方组件	新功能： <ul style="list-style-type: none"><li>用户可以在发送请求时添加任意自定义头域</li></ul> 适配第三方组件： <ul style="list-style-type: none"><li>使用 jackson-core 2.12.5 替代 jackson-core 2.11.1</li><li>使用 jackson-databind 2.12.5 替代 jackson-databind 2.11.1</li><li>使用 jackson-annotations 2.12.5 替代 jackson-annotations 2.11.1</li><li>使用 okhttp 4.9.1 替代 okhttp 4.8.0</li><li>使用 log4j2 2.16.0 替代 log4j2 2.14.1</li></ul>	是

完整的版本变更情况请参见：[ChangeLog](#)。

## 兼容性说明

- 推荐使用的JDK版本：JDK 8及以上版本。
- 旧版本（2.x）已不再维护，建议尽快升级至最新版。

## 其他使用前须知

- 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）**、**终端节点（Endpoint）**和**访问域名**等。
- 您可以先参考**OBS客户端通用示例**，了解OBS Java SDK接口调用的通用方式。
- 使用OBS客户端进行接口调用操作完成后，没有异常抛出，则表明返回值有效；如果抛出异常，则说明操作失败，此时可从**SDK自定义异常**实例中获取错误信息。
- 使用OBS客户端进行接口调用成功后，均会返回包含响应头信息的**SDK公共响应头实例**（或其子类实例）。
- 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。您可以查看**功能总览**确认区域是否支持该功能特性，或者**提交工单**寻求技术支持。

## 技术支持渠道

开发者社区提供的技术支持渠道如下：

- 开发过程中，您有任何问题可以在Github上**提交issue**，或者在**华为云对象存储服务论坛**中发帖求助。

# 2 Java SDK 接口概览(Java SDK)

## 桶相关接口

桶相关接口如下表所示。

表 2-1 桶相关接口概览

接口名	方法	功能描述	示例代码源代码
创建桶	obsClient.createBucket(CreateBucketRequest request)	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。	BucketOperationsSample
列举桶	obsClient.listBuckets(ListBucketsRequest request)	列举当前账号所有地域下符合指定条件的桶。返回结果以桶名的字典序排序。	BucketOperationsSample
删除桶	obsClient.deleteBucket(String bucketName)	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。	BucketOperationsSample
判断桶是否存在	obsClient.headBucket(String bucketName)	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。	BucketOperationsSample
获取桶元数据	obsClient.getBucketMetadata(BucketMetadataInfoRequest request)	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、冗余策略等信息。	BucketOperationsSample
设置桶ACLs	obsClient.setBucketAcl(String bucketName, AccessControlList acl)	设置指定桶的ACL策略。	BucketOperationsSample

接口名	方法	功能描述	示例代码源代码
获取桶 ACLs	obsClient.getBucketAcl(String bucketName)	获取指定桶的ACL策略。	<a href="#">BucketOperationsSample</a>
设置桶策略	obsClient.setBucketPolicy(String bucketName, String policy)	配置桶的策略，如果桶已存在策略，那么当前请求中的策略将完全覆盖桶中现存的策略。	<a href="#">BucketOperationsSample</a>
获取桶策略	obsClient.getBucketPolicy(String bucketName)	获取指定桶的桶策略。	<a href="#">BucketOperationsSample</a>
删除桶策略	obsClient.deleteBucketPolicy(String bucketName)	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。	<a href="#">BucketOperationsSample</a>
获取桶区域位置	obsClient.getBucketLocation(String bucketName)	获取桶所在的区域位置。	<a href="#">BucketOperationsSample</a>
获取桶存量信息	obsClient.getBucketStorageInfo(String bucketName)	获取桶的存量信息，包括桶已使用的空间大小以及桶包含的对象个数。	<a href="#">BucketOperationsSample</a>
设置桶配额	obsClient.setBucketQuota(String bucketName, BucketQuota bucketQuota)	设置桶的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。	<a href="#">BucketOperationsSample</a>
获取桶配额	obsClient.getBucketQuota(String bucketName)	获取指定桶的配额值。桶配额为0代表桶容量没有上限。	<a href="#">BucketOperationsSample</a>
设置桶存储类别	obsClient.setBucketStoragePolicy(String bucketName, BucketStoragePolicyConfiguration bucketStorage)	设置指定桶的存储类别。设置桶的存储类别后，如果上传对象、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。	<a href="#">BucketOperationsSample</a>
获取桶存储类别	obsClient.getBucketStoragePolicy(String bucketName)	获取指定桶的存储类别。	<a href="#">BucketOperationsSample</a>

接口名	方法	功能描述	示例代码源代码
<a href="#">设置桶清单规则</a>	obsClient.setInventoryConfiguration(SetInventoryConfigurationRequest request)	定期生成桶内对象的相关信息，保存在CSV格式的文件中，并上传到您指定的桶中，方便您管理桶内对象。指定的对象内容包括对象版本、大小、上次修改时间、存储类别、标签、加密状态等。	<a href="#">BucketOperationsSample</a>
<a href="#">获取桶清单规则</a>	obsClient.getInventoryConfiguration(GetInventoryConfigurationRequest request)	获取指定桶的清单规则，可以通过配置规则id来选择指定清单规则。	<a href="#">BucketOperationsSample</a>
<a href="#">列举桶清单规则</a>	obsClient.listInventoryConfiguration(ListInventoryConfigurationRequest request)	列举指定桶的所有清单配置，获取到的清单配置一次性返回，不分页。	<a href="#">BucketOperationsSample</a>
<a href="#">删除桶清单规则</a>	obsClient.deleteInventoryConfiguration(DeleteInventoryConfigurationRequest request)	删除指定桶的清单配置（通过清单规则id来指定确认要删除的桶清单规则）。	<a href="#">BucketOperationsSample</a>

## 并行文件系统相关接口

并行文件系统相关接口如下表所示。

表 2-2 并行文件系统相关接口概览

接口名	方法	功能描述	示例代码源代码
<a href="#">创建并行文件系统</a>	obsClient.createBucket(CreateBucketRequest request)	在当前账号按照用户指定的并行文件系统名称创建一个新并行文件系统，接口支持根据用户诉求，在创建并行文件系统的同时配置并行文件系统的区域及桶的访问权限等参数。	<a href="#">PFSBucketAndObjectOperationsSample</a>
<a href="#">列举并行文件系统</a>	obsClient.listBuckets(ListBucketsRequest request)	列举当前账号下符合指定条件的并行文件系统。返回结果以文件系统名的字典序排列。	<a href="#">PFSBucketAndObjectOperationsSample</a>
<a href="#">列举并行文件系统内对象</a>	obsClient.listObjects(final ListObjectsRequest request)	列举出指定并行文件系统里的对象。	<a href="#">PFSBucketAndObjectOperationsSample</a>

接口名	方法	功能描述	示例代码源代码
<a href="#">修改写对象</a>	obsClient.modifyObject(ModifyObjectRequest request)	将指定并行文件系统内的一个对象从指定位置起修改为其他内容。	<a href="#">PFSSBucketAndObjectOperationSample</a>
<a href="#">重命名对象</a>	obsClient.renameObject(RenameObjectRequest request)	重命名对象操作是指将指定并行文件系统内的一个对象重命名为其他对象名。	<a href="#">PFSSBucketAndObjectOperationSample</a>
<a href="#">截断对象</a>	obsClient.truncateObject(TruncateObjectRequest request)	截断对象操作是指将指定并行文件系统内的一个对象截断到指定大小。	<a href="#">PFSSBucketAndObjectOperationSample</a>

## 对象相关接口

对象相关接口如下表所示。

表 2-3 对象相关接口概览

接口名	方法	功能描述	示例代码源代码
<a href="#">流式上传</a>	obsClient.putObject(PutObjectRequest request)	通过流式上传方式将本地文件上传至OBS指定的位置，支持上传小于5GB的文件。待上传的文件可以是任何类型：文本文件、图片、视频等。	<a href="#">ObjectOperationSample</a>
<a href="#">文件上传</a>	obsClient.putObject(PutObjectRequest request)	将本地文件直接通过Internet上传至OBS指定的桶中。待上传的文件可以是任何类型：文本文件、图片、视频等。	-
<a href="#">获取上传进度</a>	PutObjectRequest.setProgressListener(ProgressListener progressListener)	获取指定对象的上传进度。	-
<a href="#">创建文件夹</a>	obsClient.putObject(PutObjectRequest request)	在已创建的桶中新建一个文件夹，从而更方便的对存储在OBS中的数据进行分类管理。	<a href="#">CreateFolderSample</a>
<a href="#">设置对象元数据</a>	obsClient setObjectMetadata(SetObjectMetadataRequest request)	在上传对象时设置对象属性。对象属性包含对象长度、对象MIME类型、对象MD5值（用于校验）、对象存储类别、对象自定义元数据。对象属性可以在多种上传方式下（流式上传、文件上传、分段上传），或复制对象时进行设置。	<a href="#">ObjectMetaSample</a>

接口名	方法	功能描述	示例代码源代码
<a href="#">初始化分段上传任务</a>	obsClient.initiateMultipartUpload(InitiateMultipartUploadRequest request)	使用分段上传方式传输数据前，必须先通知OBS初始化一个分段上传任务。该操作会返回一个OBS服务端创建的全局唯一标识（Upload ID），用于标识本次分段上传任务。	<a href="#">SimpleMultipartUploadSample</a>
<a href="#">上传段</a>	obsClient.uploadPart(UploadPartRequest request)	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。	<a href="#">SimpleMultipartUploadSample</a>
<a href="#">合并段</a>	obsClient.completeMultipartUpload(CompleteMultipartUploadRequest request)	通过分段上传任务的ID和对应已上传的段信息（包括PartNumber和ETag），合并成一个完整的对象。	<a href="#">SimpleMultipartUploadSample</a>
<a href="#">取消分段上传任务</a>	obsClient.abortMultipartUpload(AbortMultipartUploadRequest request)	通过分段上传任务的ID，取消指定桶中的分段上传任务。	-
<a href="#">列举已上传的段</a>	obsClient.listParts(ListPartsRequest request)	通过分段上传任务的ID，列举指定桶中已上传的段。	<a href="#">ConcurrentUploadPartSample</a>
<a href="#">列举分段上传任务</a>	obsClient.listMultipartUploads(ListMultipartUploadsRequest request)	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。	-
<a href="#">设置对象生命周期</a>	obsClient.putObject(PutObjectRequest request)	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。此接口设置的对象过期时间，其优先级高于桶生命周期规则。	-
<a href="#">追加上传</a>	obsClient.appendObject(AppendObjectRequest request)	对同一个对象追加数据内容。	-
<a href="#">断点续传上传</a>	obsClient.uploadFile(UploadFileRequest request)	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。	-
<a href="#">基于表单上传</a>	obsClient.createPostSignature(PostSignatureRequest request)	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。	<a href="#">PostObjectSample</a>



接口名	方法	功能描述	示例代码源代码
<a href="#">流式下载</a>	obsClient.getObject(GetObjectRequest request)	根据需要通过流式下载将存储在OBS中的指定对象下载到本地。接口返回的ObsObject实例包含对象所在的桶、对象名、对象属性、对象输入流等内容，同时可以通过操作对象输入流将对象的内容读取到本地文件或者内存中。	<a href="#">DownloadSample</a>
<a href="#">范围下载</a>	obsClient.getObject(GetObjectRequest request)	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。	-
<a href="#">获取下载进度</a>	GetObjectRequest.setProgressListener(ProgressListener progressListener)	获取指定对象的下载进度。	-
<a href="#">限定条件下载</a>	obsClient.getObject(GetObjectRequest request)	下载对象时，可以指定一个或多个限定条件，满足限定条件时则进行下载，否则返回异常码，下载对象失败。	-
<a href="#">重写响应头</a>	obsClient.getObject(GetObjectRequest request)	下载对象时，可以重写HTTP/HTTPS中部分响应头的信息：Content-Type、Content-Language、Expires、Cache-Control、Content-Disposition、Content-Encoding。	-
<a href="#">获取自定义元数据</a>	obsClient.getObject(GetObjectRequest request)	本接口可以在下载对象成功后返回对象的自定义元数据。	<a href="#">ObjectMetaSample</a>
<a href="#">恢复归档存储对象</a>	obsClient.restoreObject(RestoreObjectRequest request)	如果要下载归档存储对象，需要先将归档存储对象恢复。恢复归档存储对象的恢复选项可支持标准恢复、快速恢复。	<a href="#">RestoreObjectSample</a>
<a href="#">断点续传下载</a>	obsClient.downloadFile(DownloadFileRequest request)	对范围下载的封装和加强，解决下载大对象到本地时由于网络不稳定或程序崩溃导致下载失败的问题。	-
<a href="#">下载对象接口实现图片处理</a>	obsClient.getObject(GetObjectRequest request)	下载图片文件时，通过传入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。	-

接口名	方法	功能描述	示例代码源代码
<a href="#">临时授权方式实现图片处理</a>	obsClient.createTemporarySignature(TemporarySignatureRequest request)	通过临时授权方式传入图片处理参数，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。	-
<a href="#">设置对象元数据</a>	obsClient.setObjectMetadata(SetObjectMetadataRequest request)	对指定桶中的对象发送HEAD请求，设置对象的元数据信息。	-
<a href="#">获取对象元数据</a>	obsClient.getObjectMetadata(GetObjectMetadataRequest request)	对指定桶中的对象发送HEAD请求，获取对象的元数据信息。	-
<a href="#">设置对象ACLs</a>	obsClient.setObjectAcl(SetObjectAclRequest request)	在上传对象时设置权限控制策略，也可以通过ACL操作API接口对已存在的对象更改ACL。	<a href="#">ObjectOperationsSample</a>
<a href="#">获取对象ACLs</a>	obsClient.getObjectAcl(GetObjectAclRequest request)	通过接口获取指定桶中对象的ACL访问权限，返回信息包含指定对象的权限控制列表信息。	<a href="#">ObjectOperationsSample</a>
<a href="#">列举对象</a>	obsClient.listObjects(ListObjectsRequest request)	列举指定桶内的部分或所有对象的描述信息。还可以通过设置前缀、数量、起始位置等参数，返回符合筛选条件的对象信息。返回结果以对象名的字典序排序。	<a href="#">ListObjectsSample</a>
<a href="#">删除对象</a>	obsClient.deleteObject(DeleteObjectRequest request)	根据需要删除指定桶中的对象，节省空间和成本。	<a href="#">ObjectOperationsSample</a>
<a href="#">批量删除对象</a>	obsClient.deleteObjects(DeleteObjectsRequest deleteRequest)	根据需要批量删除指定桶中的多个对象，节省空间和成本。 批量删除对象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。	<a href="#">DeleteObjectsSample</a>
<a href="#">复制对象</a>	obsClient.copyObject(CopyObjectRequest request)	为指定桶中的对象创建一个副本。在单次操作中，可以创建最大5GB的对象副本。	<a href="#">ObjectOperationsSample</a>
<a href="#">分段复制</a>	obsClient.copyPart(CopyPartRequest request)	初始化分段上传任务后，通过分段上传任务的ID，复制段到指定桶中。	<a href="#">ConcurrentCopyPartSample</a>

接口名	方法	功能描述	示例代码源代码
<a href="#">判断对象是否存在</a>	doesObjectExist(final GetObjectMetadataRequest request)	判断对象是否存在，返回的结果中HTTP状态码为200表明对象存在，否则返回404表明对象或桶不存在。	-

## 临时授权访问相关接口

临时授权访问相关接口如下表所示。

表 2-4 临时授权访问相关接口概览

接口名	方法	功能描述	示例代码源代码
<a href="#">通过临时URL访问 OBS</a>	obsClient.createTemporarySignature(TemporarySignatureRequest request)	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，您需要指定URL的有效期来限制访客用户的访问时长。  如果您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（例如使用生成PUT请求的URL上传对象），将该URL提供给其他用户。	<a href="#">TemporarySignatureSample</a>

## 多版本控制相关接口

多版本控制相关接口如下表所示。

表 2-5 多版本控制相关接口概览

接口名	方法	功能描述	示例代码源代码
<a href="#">设置桶多版本状态</a>	obsClient.setBucketVersioning(final SetBucketVersioningRequest request)	为指定桶设置多版本状态。在一个桶中保留对象的多个版本，可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。	<a href="#">BucketOperationsSample</a>

接口名	方法	功能描述	示例代码源代码
获取桶多版本状态	obsClient.getBucketVersioning(final BaseBucketRequest request)	获取指定桶的多版本状态。	<a href="#">BucketOperationSample</a>
获取多版本对象	obsClient.getObject(GetObjectRequest request)	获取指定多版本对象。	-
复制多版本对象	obsClient.copyObject(CopyObjectRequest request)	为指定桶中的多版本对象创建一个副本。在单次操作中，可以创建最大5GB的对象副本。	-
恢复多版本归档存储对象	obsClient.restoreObject(RestoreObjectRequest request)	通过接口传入版本号，恢复多版本归档存储对象。 如果要下载归档存储对象，需要先将归档存储对象恢复。恢复归档存储对象的恢复选项可支持标准恢复、快速恢复。	-
列举多版本对象	obsClient.listVersions(ListVersionsRequest request)	列举指定桶内的部分或所有多版本对象的描述信息。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多版本对象信息。返回结果以多版本对象名的字典序排序。	<a href="#">ListVersionsSample</a>
设置多版本对象权限	obsClient.setObjectAcl(SetObjectAclRequest request)	在上传多版本对象时，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。	-
获取多版本对象权限	obsClient.getObjectAcl(GetObjectAclRequest request)	获取指定桶的获取多版本对象权限。	-
删除多版本对象	obsClient.deleteObject(DeleteObjectRequest request)	根据需要删除指定桶中的多版本对象，节省空间和成本。	-
批量删除多版本对象	obsClient.deleteObjects(DeleteObjectsRequest deleteRequest)	根据需要批量删除指定桶中的多个多版本对象，节省空间和成本。 批量删除对象用于将一个桶内的部分多版本对象一次性删除，删除后不可恢复。批量删除多版本对象要求返回结果里包含每个多版本对象的删除结果。	<a href="#">ListVersionsSample</a>

## 生命周期管理相关接口

生命周期管理相关接口如下表所示。

表 2-6 生命周期管理相关接口概览

接口名	方法	功能描述	示例代码源代码
设置生命周期规则	obsClient.setBucketLifecycle(final SetBucketLifecycleRequest request)	为指定桶设置生命周期规则，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。	BucketOperationsSample
获取生命周期规则	obsClient.getBucketLifecycle(final BaseBucketRequest request)	获取指定桶的生命周期规则。	BucketOperationsSample
删除生命周期规则	obsClient.deleteBucketLifecycle(final BaseBucketRequest request)	删除指定桶的生命周期规则。	BucketOperationsSample

## 跨域资源共享相关接口

跨域资源共享相关接口如下表所示。

表 2-7 跨域资源共享相关接口概览

接口名	方法	功能描述	示例代码源代码
设置跨域资源共享规则	obsClient.setBucketCors(final SetBucketCorsRequest request)	设置桶的跨域资源共享规则，允许客户端浏览器进行跨域请求。设置成功后，如果原规则存在则覆盖原规则。	BucketOperationsSample
获取跨域资源共享规则	obsClient.getBucketCors(final BaseBucketRequest request)	获取指定桶的跨域资源共享规则。	BucketOperationsSample
删除跨域资源共享规则	obsClient.deleteBucketCors(final BaseBucketRequest request)	删除指定桶的跨域资源共享规则。	BucketOperationsSample

## 桶日志相关接口

设置访问日志相关接口如下表所示。

表 2-8 桶日志相关接口概览

接口名	方法	功能描述	示例代码源代码
<a href="#">设置桶日志规则</a>	obsClient.setBucketLogging(final SetBucketLoggingRequest request)	为指定桶打开桶日志功能，并配置日志存放的目标桶。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 to 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。	<a href="#">BucketOperationsSample</a>
<a href="#">获取桶日志规则</a>	obsClient.getBucketLogging(final BaseBucketRequest request)	获取指定桶的日志配置。	<a href="#">BucketOperationsSample</a>

## 静态网站托管相关接口

静态网站托管相关接口如下表所示。

表 2-9 静态网站托管相关接口概览

接口名	方法	功能描述	示例代码源代码
<a href="#">网站文件托管</a>	<ol style="list-style-type: none"> <li>obsClient.putObject(PutObjectRequest request)</li> <li>obsClient.setObjectAcl(SetObjectAclRequest acl)</li> </ol>	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。	-
<a href="#">设置托管配置</a>	obsClient.setBucketWebsite(final SetBucketWebsiteRequest request)	为指定桶设置网站配置信息。	<a href="#">BucketOperationsSample</a>

接口名	方法	功能描述	示例代码源代码
<a href="#">获取托管配置</a>	obsClient.getBucketWebsite(final BaseBucketRequest request)	获取指定桶的网站配置信息。	<a href="#">BucketOperationsSample</a>
<a href="#">删除托管配置</a>	obsClient.deleteBucketWebsite(final BaseBucketRequest request)	删除指定桶的网站配置。	<a href="#">BucketOperationsSample</a>

## 标签管理相关接口

标签管理相关接口如下表所示。

表 2-10 标签管理相关接口概览

接口名	方法	功能描述	示例代码源代码
<a href="#">设置桶标签</a>	obsClient.setBucketTagging(final SetBucketTaggingRequest request)	为桶添加标签，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。	<a href="#">BucketOperationsSample</a>
<a href="#">获取桶标签</a>	obsClient.getBucketTagging(final BaseBucketRequest request)	获取指定桶的标签。	<a href="#">BucketOperationsSample</a>
<a href="#">删除桶标签</a>	obsClient.deleteBucketTagging(final BaseBucketRequest request)	删除指定桶的标签。	<a href="#">BucketOperationsSample</a>

# 3 使用前准备(Java SDK)

在使用Java SDK访问对象存储服务OBS之前，您需要先完成服务环境的准备和开发环境的准备。服务环境准备包括准备账号和访问密钥，是使用SDK与OBS云服务交互的必要条件。开发环境准备是指为了您能顺利完成SDK的安装、完成基于SDK的代码开发与运行，需要提前在本地完成开发环境的搭建，比如下载安装依赖软件、安装开发工具等。

## 准备华为账号

使用OBS之前您必须要有一个华为账号。申请华为账号步骤详见：[注册华为账号并开通华为云](#)。

## 准备访问密钥

OBS通过用户账号中的AK和SK进行签名验证，确保通过授权的账号才能访问指定的OBS资源。获取访问密钥前，请确保访问OBS的IAM子用户已开启编程访问，开启方式详见[修改或查看IAM用户信息](#)。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。

- 永久访问密钥：
  - a. 登录OBS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”。
  - e. 在弹出的“新增访问密钥”对话框中，输入登录密码和对应验证码。



### 📖 说明

- 用户如果未绑定邮箱和手机，则只需输入登录密码。
  - 用户如果同时绑定了邮箱和手机，可以选择其中一种方式进行验证。
- f. 单击“确定”。
  - g. 在弹出的“下载确认”提示框中，单击“确定”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”文件既可获取到访问密钥（AK和SK）。

### 📖 说明

- 每个用户最多可创建两个有效的访问密钥。
  - 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。

---

## 准备开发环境

- 从[Oracle官网](#)下载并安装推荐使用的JDK版本。推荐使用的JDK版本：JDK 8及以上版本。
- 从[Eclipse官网](#)下载并安装Eclipse IDE for Java Developers最新版本。

# 4 下载与安装 SDK(Java SDK)

## 下载 SDK

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

### 📖 说明

如需采用Maven依赖方式获取SDK，请参见[安装SDK](#)章节。

## 安装方式概览

如[表1](#)所示，Java SDK以下安装方式。

表 4-1 Java SDK 安装方式概览

序号	方式
1	<a href="#">使用Maven中央仓库和Maven工程下载安装OBS Java SDK</a>
2	<a href="#">使用Gradle中央仓库和Gradle工程下载安装OBS Java SDK</a>
3	<a href="#">使用镜像库下载安装OBS Java SDK</a>
4	<a href="#">自行编译 Jar 包</a>

## 方式一：使用 Maven 中央仓库和 Maven 工程下载安装 OBS Java SDK

使用本方式安装SDK前，请确保Java环境和Maven环境正确配置并能正常使用。

**步骤1** 访问Maven中央仓库获取OBS Java SDK版本号。获取Bundle版本请访问[链接](#)，获取普通版本请访问[链接](#)，建议使用最新版本的SDK。例如，此处选择Bundle 3.23.9版本。

### 📖 说明

- Bundle版本与普通版本源代码相同，区别在于 Bundle 版将所有三方依赖打包并重定向至包内，不再依赖外部三方包，可避免因依赖冲突导致的问题，相应的 Bundle 版 SDK 占用的空间也更大（7M+）。
- 在使用普通版本的过程中遇到依赖冲突问题时，可参考 [依赖缺失和依赖冲突的解决\(Java SDK\)](#) 解决。

**步骤2** 打开Maven工程的pom.xml文件，在“<dependencies>”节点中加入依赖配置。

如果您选择使用Bundle版本，请将以下配置添加到“<dependencies>”节点中，并将“bundleVersionNumber”修改为步骤**步骤1**中获取的Java SDK版本号：

```
<dependency>
  <groupId>com.huaweicloud</groupId>
  <artifactId>esdk-obs-java-bundle</artifactId>
  <version>bundleVersionNumber</version>
</dependency>
```

例如您如果选择使用Bundle 3.23.9版本，那么您的pom.xml文件应该插入以下配置：

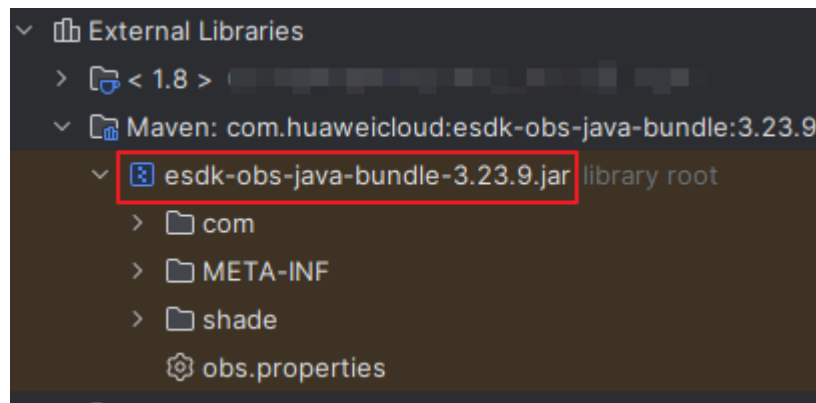
```
<dependency>
  <groupId>com.huaweicloud</groupId>
  <artifactId>esdk-obs-java-bundle</artifactId>
  <version>3.23.9</version>
</dependency>
```

如果您选择使用普通版本，请将以下配置添加到“<dependencies>”节点，并将“VersionNumber”修改为为步骤**步骤1**中获取的Java SDK版本号：

```
<dependency>
  <groupId>com.huaweicloud</groupId>
  <artifactId>esdk-obs-java</artifactId>
  <version>versionNumber</version>
</dependency>
```

**步骤3** 刷新Maven配置，即Reload All Maven Projects。

**步骤4** 如果在您的依赖路径下，下载了Java SDK的jar包，则说明Java SDK安装成功。如果未下载jar包，请检查是否正确配置了pom.xml文件，版本号是否正确替换。如仍未解决，请[提交工单](#)寻求技术支持。



----结束

## 方式二：使用 Gradle 中央仓库和 Gradle 工程下载安装 OBS Java SDK

使用本方式安装SDK前，请确保Java环境和Gradle环境正确配置并能正常使用。

**步骤1** 获取OBS Java SDK版本号。获取Bundle版本请访问[链接](#)，获取普通版本请访问[链接](#)，建议使用最新版本的SDK。例如，此处选择Bundle 3.23.9版本。

### 📖 说明

- Bundle版本与普通版本源代码相同，区别在于 Bundle 版将所有三方依赖打包并重定向至包内，不再依赖外部三方包，可避免因依赖冲突导致的问题，相应的 Bundle 版 SDK 占用的空间也更大（7M+）。
- 在使用普通版本的过程中遇到依赖冲突问题时，可参考 [依赖缺失和依赖冲突的解决\(Java SDK\)](#) 解决。

**步骤2** 打开Gradle工程的build.gradle文件，在“dependencies”中加入依赖配置。

如果您选择使用Bundle版本，请将以下配置添加到“dependencies”节点中，并将“bundleVersionNumber”修改为步骤**步骤1**中获取的Java SDK版本号：

```
api 'com.huaweicloud:esdk-obs-java-bundle:bundleVersionNumber'
```

例如您如果选择使用Bundle 3.23.9版本，那么您的pom.xml文件应该插入以下配置：

```
api 'com.huaweicloud:esdk-obs-java-bundle:3.23.9'
```

如果您选择使用普通版本，请将以下配置添加到“<dependencies>”节点，并将“VersionNumber”修改为步骤**步骤1**中获取的Java SDK版本号：

```
api 'com.huaweicloud:esdk-obs-java:versionNumber'
```

**步骤3** 刷新Gradle配置，即Reload All Gradle Projects。

**步骤4** 如果在您的依赖路径下，下载了Java SDK的jar包，则说明Java SDK安装成功。如果未生成jar包，请检查是否正确配置了build.gradle文件，版本号是否正确替换。如仍未解决，请[提交工单](#)寻求技术支持。

----结束

## 方式三：使用镜像库下载安装 OBS Java SDK

使用本方式安装SDK前，请确保Java环境和Maven环境正确配置并能正常使用，详细步骤如下：

**步骤1** 使用浏览器打开华为开源镜像站：<https://mirrors.huaweicloud.com>。

**步骤2** 从打开站点的网页中，找到“HuaweiCloud SDK”面板并单击进入依赖管理工具配置说明页面。

**步骤3** 在依赖管理工具配置说明页面，单击右上方的“下载配置文件”，下载并保存Maven配置文件。

### HuaweiCloud SDK

✔ 享受3~5MB/s CDN加速下载

#### 简介

华为云SDK

#### 配置方法

##### 依赖管理工具

选择依赖管理工具

Maven

[下载配置文件](#)

[新手指引](#)

##### 准备工作

**步骤4** 使用下载好的Maven配置文件（ settings.xml ）替换本地Maven的全局配置文件，例如 windows系统中该文件通常位于 “C:\Users\**<administrator>**\.m2” 下。

**步骤5** 打开Maven工程的pom.xml文件，在 “<dependencies>” 节点中加入依赖配置。

如果您选择使用Bundle版本，请将以下配置添加到 “<dependencies>” 节点中，并将 “bundleVersionNumber” 修改为您正在使用的Java SDK版本号：

```
<dependency>
  <groupId>com.huaweicloud</groupId>
  <artifactId>esdk-obs-java-bundle</artifactId>
  <version>bundleVersionNumber</version>
</dependency>
```

例如您如果选择使用Bundle 3.23.9版本，那么您的pom.xml文件应该插入以下配置：

```
<dependency>
  <groupId>com.huaweicloud</groupId>
  <artifactId>esdk-obs-java-bundle</artifactId>
  <version>3.23.9</version>
</dependency>
```

如果您选择使用普通版本，请将以下配置添加到 “<dependencies>” 节点，并将 “VersionNumber” 修改为您正在使用的Java SDK版本号：

```
<dependency>
  <groupId>com.huaweicloud</groupId>
  <artifactId>esdk-obs-java</artifactId>
  <version>versionNumber</version>
</dependency>
```

**步骤6** 运行Maven命令（如： mvn package ）下载SDK。

----结束

## 方式四：自行编译 Jar 包

您可以下载Java SDK源码，然后自行编译 Jar 包。使用本方式安装SDK前，请确保Java环境和Maven环境正确配置并能正常使用，详细步骤如下：

**步骤1** **下载**SDK源码并解压。

**步骤2** 通过命令行进入源码解压目录。

**步骤3** 运行如下命令

```
mvn clean package -Dmaven.test.skip=true -f pom-java.xml
```

**步骤4** 构建产物位于解压目录中的target目录下。

**步骤5** 将生成的jar包放置于本地工程的依赖路径。

----结束

# 5 快速入门(Java SDK)

本节介绍在完成SDK的下载和安装后，如何快速上手使用OBS的基础功能，完成创建桶、上传下载对象、列举对象和删除对象。

## 入门准备

使用SDK管理OBS资源，您需要先完成以下几项准备工作：

1. [使用前须知\(Java SDK\)](#)：了解并选择合适的SDK版本。
2. [使用前准备\(Java SDK\)](#)：完成服务环境和开发环境准备。
3. [下载与安装SDK\(Java SDK\)](#)：下载SDK并完成安装。

## 创建桶

以下代码展示了创建一个名为examplebucket的桶，设置桶所在区域为华北-北京四，设置桶权限为私有，存储类别为标准存储，存储冗余类别为单AZ。了解更多请参见[创建桶\(Java SDK\)](#)。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
import com.obs.services.model.AvailableZoneEnum;
import com.obs.services.model.CreateBucketRequest;
import com.obs.services.model.ObsBucket;
import com.obs.services.model.StorageClassEnum;

public class CreateBucket001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");

        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";

        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
```

```
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    CreateBucketRequest request = new CreateBucketRequest();
    // 示例桶名
    String exampleBucket = "examplebucket";
    // 示例桶区域位置
    String exampleLocation = "cn-north-4";
    request.setBucketName(exampleBucket);
    // 设置桶访问权限为私有读写，默认也是私有读写
    request.setAcl(AccessControlList.REST_CANNED_PRIVATE);
    // 设置桶的存储类别为标准存储
    request.setBucketStorageClass(StorageClassEnum.STANDARD);
    // 设置桶区域位置(以区域为华北-北京四为例)，location 需要与 endpoint的位置信息一致
    request.setLocation(exampleLocation);
    // 指定创建多AZ桶，如果不设置，默认创建单AZ桶
    request.setAvailableZone(AvailableZoneEnum.MULTI_AZ);
    // 创建桶
    ObsBucket bucket = obsClient.createBucket(request);
    // 创建桶成功
    System.out.println("CreateBucket successfully");
    System.out.println("RequestId:" + bucket.getRequestId());

} catch (ObsException e) {
    System.out.println("CreateBucket failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code: " + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message: " + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
} catch (Exception e) {
    System.out.println("CreateBucket failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 上传对象

以下代码展示了向名为examplebucket的桶中上传2个对象，一个对象本地文件名为localfile，上传到桶中是名为objectkey，另一个对象本地文件名为localfile2，上传到桶中是名为objectkey2。了解更多请参见[上传对象\(Java SDK\)](#)。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.PutObjectRequest;
import java.io.File;
public class PutObject004 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
    }
}
```

```
// endpoint填写桶所在的endpoint, 此处以华北-北京四为例, 其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint, 也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 文件上传
    // localfile 为待上传的本地文件路径, 需要指定到具体的文件名
    PutObjectRequest request = new PutObjectRequest();
    request.setBucketName("examplebucket");
    request.setObjectKey("objectkey");
    request.setFile(new File("localfile"));
    obsClient.putObject(request);
    System.out.println("putObject successfully");
} catch (ObsException e) {
    System.out.println("putObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("putObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 下载对象

以下代码展示了下载examplebucket桶中名为objectname的对象。了解更多请参见[下载对象\(Java SDK\)](#)。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObsObject;
import java.io.ByteArrayOutputStream;
import java.io.InputStream;
public class GetObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK, 也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken, 也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint, 此处以华北-北京四为例, 其他地区请按实际情况填写。查看桶所在的endpoint请参见: https://support.huaweicloud.com/usermanual-obs/obs_03_0312.html。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint, 也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
```



```
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 流式下载
    ObsObject obsObject = obsClient.getObject("examplebucket", "objectname");
    // 读取对象内容
    System.out.println("Object content:");
    InputStream input = obsObject.getObjectContent();
    byte[] b = new byte[1024];
    ByteArrayOutputStream bos = new ByteArrayOutputStream();
    int len;
    while ((len = input.read(b)) != -1) {
        bos.write(b, 0, len);
    }
    System.out.println("getObjectContent successfully");
    System.out.println(new String(bos.toByteArray()));
    bos.close();
    input.close();
} catch (ObsException e) {
    System.out.println("getObjectContent failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getObjectContent failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 列举对象

以下代码展示了列举名为examplebucket桶中的对象。了解更多请参见[列举对象\(Java SDK\)](#)。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObjectListing;
import com.obs.services.model.ObsObject;
public class ListObjects001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK, 也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken, 也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint, 此处以华北-北京四为例, 其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint, 也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
```

```
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 简单列举
    ObjectListing result = obsClient.listObjects("examplebucket");
    for (ObsObject obsObject : result.getObjects()) {
        System.out.println("listObjects successfully");
        System.out.println("ObjectKey:" + obsObject.getObjectKey());
        System.out.println("Owner:" + obsObject.getOwner());
    }
} catch (ObsException e) {
    System.out.println("listObjects failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("listObjects failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 删除对象

以下代码展示了删除examplebucket桶中名为objectname的对象。了解更多请参见[删除对象\(Java SDK\)](#)。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class DeleteObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK, 也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken, 也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint, 此处以华北-北京四为例, 其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint, 也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 删除单个对象
            obsClient.deleteObject("examplebucket", "objectname");
            System.out.println("deleteObject successfully");
        } catch (ObsException e) {
            System.out.println("deleteObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
        }
    }
}
```

```
// 请求失败,打印详细错误信息
System.out.println("Error Message:" + e.getMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("deleteObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## OBS 客户端通用示例

以下代码展示了使用OBS客户端的通用方式:

```
// 您的工程中可以只保留一个全局的ObsClient实例
// ObsClient是线程安全的,可在并发场景下使用
ObsClient obsClient = null;
try
{
    String endPoint = "https://your-endpoint";
    // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;本示例以ak和sk保存在环境变量中为例,运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca\_01\_0003.html
    String ak = System.getenv("ACCESS_KEY_ID");
    String sk = System.getenv("SECRET_ACCESS_KEY_ID");
    // 创建ObsClient实例
    obsClient = new ObsClient(ak, sk, endPoint);
    // 调用接口进行操作,例如上传对象
    HeaderResponse response = obsClient.putObject("bucketname", "objectname", new File("localfile")); // localfile为待上传的本地文件路径,需要指定到具体的文件名
    System.out.println(response);
}
catch (ObsException e)
{
    System.out.println("HTTP Code: " + e.getResponseCode());
    System.out.println("Error Code:" + e.getErrorCode());
    System.out.println("Error Message: " + e.getMessage());

    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    Map<String, String> headers = e.getResponseHeaders();// 遍历Map的entry,打印所有报错相关头域
    if(headers != null){
        for (Map.Entry<String, String> header : headers.entrySet()) {
            if(header.getKey().contains("error")){
                System.out.println(header.getKey()+":"+header.getValue());
            }
        }
    }
    e.printStackTrace();
}finally{
    // 关闭ObsClient实例,如果是全局ObsClient实例,可以不在每个方法调用完成后关闭
    // ObsClient在调用ObsClient.close方法关闭后不能再次使用
    if(obsClient != null){
        try
        {
            // obsClient.close();
        }
        catch (IOException e)
        {
        }
    }
}
```

# 6 初始化(Java SDK)

## 6.1 初始化概述(Java SDK)

完成SDK安装之后，您还需要进行初始化工作，初始化工作主要包括两方面，一方面客户端的创建和配置，另一方面是SDK日志配置。

表 6-1 初始化概述

初始化任务	子任务	是否必选	说明
创建并配置客户端	创建客户端	是	OBS客户端是使用Java SDK访问OBS服务的必备媒介，它为调用者提供一系列与OBS服务进行交互的接口，使用OBS Java SDK向OBS发起请求，您需要创建并初始化一个客户端，即一个ObsClient实例。
	配置访问密钥	是	基于安全需要，您需要为客户端配置访问凭证以便服务端校验请求的合法性。
	配置终端节点及其他选项	终端节点必选，其余配置项可选	<ul style="list-style-type: none"><li>为了快速创建和检索到您的资源，需要配置客户端的终端节点，即告知服务端资源所在的区域。</li><li>除了访问密钥和终端节点外，您可以根据自身使用场景，为客户端配置其他选项。</li></ul>
配置SDK日志	配置日志等级	否	OBS Java SDK基于Apache Log4j2开源库提供了日志功能，Log4j2框架按照优先级将日志进行了分级处理，您可以配置存储某个级别以上的日志，自行决定存储日志的范围。 OBS Java SDK默认将WARN及WARN以上级别的日志保存到本地。
	配置日志存放路径	否	配置日志在本地的存储路径。

## 6.2 创建并配置客户端(Java SDK)

### 操作场景

本章节指导客户创建并配置客户端，该任务是使用SDK与OBS交互的必备前置任务。如果您已有客户端，请跳过该节，直接参考[Java SDK接口概览\(Java SDK\)](#)获取接口信息。

### 前提条件

初始化客户端之前，您需要先完成以下几项准备工作：

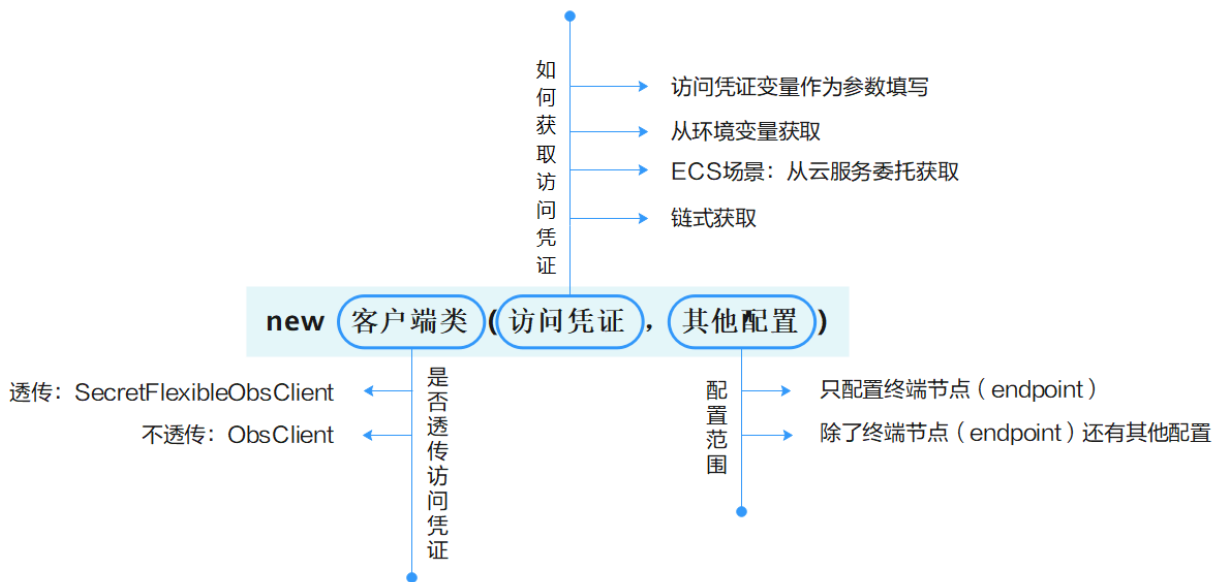
1. [使用前须知\(Java SDK\)](#)：了解并选择合适的SDK版本。
2. [使用前准备\(Java SDK\)](#)：完成服务环境和开发环境准备。
3. [下载与安装SDK\(Java SDK\)](#)：下载SDK并完成安装。

### 注意事项

- 客户端在创建时必须同步完成客户端配置，配置机会仅有一次，创建完成后，不能对客户端进行二次配置。
- 在使用临时aksk时，aksk会有过期时间，可调用 `ObsClient.refresh("yourAccessKey", "yourSecretKey", "yourSecurityToken")`刷新ObsClient的aksk，不必重新创建ObsClient。
- 建议整个代码工程全局使用一个ObsClient客户端，只在程序初始化时创建一次，因为创建多个ObsClient客户端在高并发场景下会影响性能。
- ObsClient是线程安全的，可在并发场景下使用。
- ObsClient在调用ObsClient.close方法关闭后不能再次使用，保证全局使用一个ObsClient客户端的情况下，不建议主动关闭ObsClient客户端。

## 配置概览

图 6-1 创建并配置客户端概览



如图 6-1 所示，请根据需要选择客户端类、选择获取访问密钥的方法，以及进行其他配置：

表 6-2 创建并配置客户端概览

选择项	可选项	说明	示例代码
客户端类	不透传密钥的客户端类	如果您只想配置一次访问凭证，不想在每一次调用OBS接口时都为客户端设置一次访问密钥，请使用普通客户端类ObsClient创建客户端。	<a href="#">ObsClient示例代码</a>
	透传密钥的客户端类	如果您需要在每次调用接口时，都为客户端设置一次访问凭证，OBS Java SDK提供了直接在接口方法中透传AK和SK的OBS客户端类 SecretFlexibleObsClient，SecretFlexibleObsClient继承自 ObsClient。	<a href="#">SecretFlexibleObsClient示例代码</a>

选择项	可选项	说明	示例代码
访问凭证获取方式	从变量传入	创建OBS客户端实例，并使用变量传入方式的访问凭证。	<ul style="list-style-type: none"> <li>访问凭证变量直接作为参数示例代码</li> <li>BasicObsCredentialsProvider示例代码</li> </ul>
	从环境变量获取	创建OBS客户端实例，并使用EnvironmentVariableObsCredentialsProvider从环境变量获取访问凭证。	EnvironmentVariableObsCredentialsProvider示例代码
	ECS场景从云服务委托获取	创建OBS客户端实例，并使用EcsObsCredentialsProvider从ECS服务器自动获取临时访问密钥，并且访问凭证会定期自动刷新。 <b>须知</b> <ol style="list-style-type: none"> <li>使用EcsObsCredentialsProvider的前提，应用程序须运行在ECS服务器上，并且ECS绑定了IAM对ECS的云服务委托（该委托拥有OBS权限）。</li> <li>请确保服务端和应用程序部署所在环境的UTC时间一致，否则可能会导致临时访问密钥无法及时刷新。</li> <li>使用该方式创建客户端时，SDK会请求固定IP（169.254.169.254）的API获取临时AKSK，具体请参见在<a href="#">ECS上获取Security Key</a>。</li> </ol>	EnvironmentVariableObsCredentialsProvider示例代码
	链式获取	以链式的方式从预定义方式列表中搜索访问密钥，以第一组成功获取到的访问密钥创建OBS客户端。 当前顺序是默认先从环境变量，再从ECS服务器委托中进行搜索，暂不支持自定义获取方式和顺序。	OBSCredentialsProviderChain示例代码
其他配置	只配置终端节点（endpoint）	除了访问凭证外，只配置中终端节点（endpoint）参数。	除访问凭证外只配置endpoint示例代码
	除了配置终端节点（endpoint）还有其他配置	除了配置终端节点（endpoint）还需要配置HTTP代理、Socket层传输数据超时时间等参数，您可通过ObsConfiguration配置类对ObsClient进行配置，支持的配置项参见 <a href="#">表6-3</a> 。	ObsConfiguration示例代码

## 示例代码

- **使用不透传访问凭证的客户端类，直接配置访问凭证，除访问凭证外只配置终端节点（endpoint）的示例代码**
  - 永久访问密钥（AK/SK）创建OBS客户端的代码如下：

```
// Endpoint以北京四为例，其他地区请按实际情况填写。  
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";  
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量  
// 中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在  
// 本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca_01_0003.html  
String ak = System.getenv("ACCESS_KEY_ID");  
String sk = System.getenv("SECRET_ACCESS_KEY_ID");  
// 创建ObsClient实例  
ObsClient obsClient = new ObsClient(ak, sk, endPoint);  
  
// 使用访问OBS  
  
// 关闭obsClient  
obsClient.close();
```
  - 临时访问密钥（AK/SK/SecurityToken）创建OBS客户端的代码如下：

```
// Endpoint以北京四为例，其他地区请按实际情况填写。  
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";  
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量  
// 中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在  
// 本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca_01_0003.html  
String ak = System.getenv("ACCESS_KEY_ID");  
String sk = System.getenv("SECRET_ACCESS_KEY_ID");  
String securityToken = System.getenv("SECRET_TOKEN");  
// 创建ObsClient实例  
ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);  
  
// 使用访问OBS  
  
// 关闭obsClient  
obsClient.close();
```
  - BasicObsCredentialsProvider示例代码

```
// Endpoint以北京四为例，其他地区请按实际情况填写。  
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";  
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量  
// 中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在  
// 本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca_01_0003.html  
String ak = System.getenv("ACCESS_KEY_ID");  
String sk = System.getenv("SECRET_ACCESS_KEY_ID");  
// 创建ObsClient实例  
ObsClient obsClient = new ObsClient(new BasicObsCredentialsProvider(ak, sk), endPoint);  
  
// 使用访问OBS  
  
// 关闭obsClient  
obsClient.close();
```
- **透传密钥客户端类（SecretFlexibleObsClient）示例代码**

```
// Endpoint以北京四为例，其他地区请按实际情况填写。  
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";  
// 创建ObsConfiguration配置类实例  
ObsConfiguration config = new ObsConfiguration();  
config.setEndPoint(endPoint);  
  
// 创建SecretFlexibleObsClient实例  
SecretFlexibleObsClient obsClient = new SecretFlexibleObsClient(config);  
// 使用访问OBS
```



```
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca\_01\_0003.html  
String ak1 = System.getenv("ACCESS_KEY_ID");  
String sk1 = System.getenv("SECRET_ACCESS_KEY_ID");  
obsClient.listBuckets(ak1, sk1);
```

```
String ak2 = System.getenv("ACCESS_KEY_ID");  
String sk2 = System.getenv("SECRET_ACCESS_KEY_ID");  
obsClient.listBuckets(ak2, sk2);
```

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

- **从环境变量获取访问凭证（EnvironmentVariableObsCredentialsProvider）示例代码**

使用该方法需要在环境变量中定义OBS\_ACCESS\_KEY\_ID和OBS\_SECRET\_ACCESS\_KEY分别代表永久的AK和SK。

// Endpoint以北京四为例，其他地区请按实际情况填写。

```
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
```

// 创建ObsClient实例

```
ObsClient obsClient = new ObsClient(new EnvironmentVariableObsCredentialsProvider(), endPoint);
```

// 使用访问OBS

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

- **ECS场景从云服务委托获取访问凭证（EcsObsCredentialsProvider）示例代码**

// Endpoint以北京四为例，其他地区请按实际情况填写。

```
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
```

// 创建ObsClient实例

```
ObsClient obsClient = new ObsClient(new EcsObsCredentialsProvider(), endPoint);
```

// 使用访问OBS

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

- **链式获取访问凭证（OBSCredentialsProviderChain）示例代码**

// Endpoint以北京四为例，其他地区请按实际情况填写。

```
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
```

// 创建ObsClient实例

```
ObsClient obsClient = new ObsClient(new OBSCredentialsProviderChain(), endPoint);
```

// 使用访问OBS

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

- **使用ObsConfiguration配置参数示例代码**

- 配置KeyManagerFactory，配置后可以在本地保存证书，并校验服务端返回证书是否正确。

// Endpoint以北京四为例，其他地区请按实际情况填写。

```
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
```

// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量ACCESS\_KEY\_ID和SECRET\_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/usermanual-ca/ca\\_01\\_0003.html](https://support.huaweicloud.com/usermanual-ca/ca_01_0003.html)

```
String ak = System.getenv("ACCESS_KEY_ID");
```

```
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
```

```
String jksPassword = "you-jks-password";
```

```
String jksPath = "/path/to/your/keystore/file";
```

```
KeyStore ks = KeyStore.getInstance("JKS");
```

```
char[] passArray = jksPassword.toCharArray();
FileInputStream inputStream = new FileInputStream(jksPath);
ks.load(inputStream, passArray);
KeyManagerFactory kmf =
KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());
kmf.init(ks, passArray);

String trustJKSPassword = "you-trustJKS-password";
String trustJKSPath = "/path/to/your/trustKeyStore/file";
KeyStore trustKeyStore = KeyStore.getInstance("JKS");
char[] trustPassArray = trustJKSPassword.toCharArray();
FileInputStream trustInputStream = new FileInputStream(trustJKSPath);
trustKeyStore.load(trustInputStream, trustPassArray);
TrustManagerFactory tmf =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
tmf.init(trustKeyStore);

ObsConfiguration config = new ObsConfiguration();
config.setEndPoint(endPoint);
config.setKeyManagerFactory(kmf);
config.setTrustManagerFactory(tmf);

ObsClient obsClient = new ObsClient(ak, sk, config);
```

### 📖 说明

本地证书保存格式应为 jks 格式，可以运行以下命令调用 Java 自带的 keytool 工具将 cer 证书转换为 jks 证书。

```
keytool -import -file your-cer-file.cer -keystore your-keystore-file.jks
```

- 配置 HTTP 代理，配置后 SDK 将会通过的代理访问服务端。

```
// Endpoint以北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca_01_0003.html
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY_ID");

String proxyUrl = "proxy.com";
int proxyPort = 8080;
String proxyUser = "userName";
String proxyPassword = "password";
ObsConfiguration config = new ObsConfiguration();
config.setEndPoint(endPoint);
config.setHttpProxy(proxyUrl, proxyPort, proxyUser, proxyPassword);
ObsClient obsClient = new ObsClient(ak, sk, config);
```

## 配置自定义域名访问 obs

使用自定义的域名，必须在工信部完成备案，具体配置步骤请参考：[配置自定义域名](#)。

```
// 在控制台绑定成功的自定义域名。
String endPoint = "http://your-domain";
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca_01_0003.html
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY_ID");

// 创建ObsConfiguration配置类实例
ObsConfiguration config = new ObsConfiguration();
config.setEndPoint(endPoint);
```

```

config.setCname(true);

// 创建ObsClient实例
ObsClient obsClient = new ObsClient(ak, sk, config);

// 使用Provider创建ObsClient实例
// ObsClient obsClient = new ObsClient(new EnvironmentVariableObsCredentialsProvider(), config);
// ObsClient obsClient = new ObsClient(new EcsObsCredentialsProvider(), config);

// 使用访问OBS

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

## ObsConfiguration 类支持配置的参数列表

### 说明

- 如需提高文件上传下载性能，在网络带宽满足的情况下，可对socketWriteBufferSize, sockeReadBufferSize, readBufferSize, writeBufferSize四个参数进行调优。
- 如网络状况不佳，建议增大connectionTimeout和socketTimeout的值。

表 6-3 ObsConfiguration 类支持配置的参数列表

参数	描述	方法	建议值
connectionTimeout	<b>参数解释：</b> 建立HTTP/HTTPS连接的超时时间。 <b>默认取值：</b> 60000，单位：毫秒。	ObsConfiguration.setConnectionTimeout	[10000, 60000]
socketTimeout	<b>参数解释：</b> Socket层传输数据的超时时间（单位：毫秒）。 <b>默认取值：</b> 60000，单位：毫秒。	ObsConfiguration.setSocketTimeout	[10000, 60000]
idleConnectionTime	<b>参数解释：</b> 如果空闲时间超过此参数的设定值，则关闭连接。 <b>默认取值：</b> 30000，单位：毫秒。	ObsConfiguration.setIdleConnectionTime	根据实际情况设置
maxIdleConnections	<b>参数解释：</b> 连接池中最大空闲连接数。 <b>默认取值：</b> 1000，单位：个。	ObsConfiguration.setMaxIdleConnections	根据实际情况设置
maxConnections	<b>参数解释：</b> 最大允许的HTTP并发请求数。 <b>默认取值：</b> 1000，单位：个。	ObsConfiguration.setMaxConnections	默认

参数	描述	方法	建议值
maxErrorRetry	<p><b>参数解释:</b> 请求失败（请求异常、服务端报500或503错误等）后最大的重试次数。</p> <p><b>约束限制:</b> 该参数对于上传对象和下载对象接口时，当上传或下载已经进入数据流处理阶段后产生异常中断，此时将不会重试。</p> <p><b>默认取值:</b> 3，单位：次。</p>	ObsConfiguration.setMaxErrorRetry	[0, 5]
endPoint	<p><b>参数解释:</b> 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）</p> <p><b>约束限制:</b> 出于DNS解析性能和OBS服务可靠性的考虑，不允许将endPoint设置为IP，必须使用域名访问OBS服务。</p> <p><b>默认取值:</b> 无</p>	ObsConfiguration.setEndPoint	根据实际情况设置
httpProxy	<p><b>参数解释:</b> HTTP代理配置。</p> <p><b>默认取值:</b> 默认为空</p>	ObsConfiguration.setHttpProxy	根据实际情况设置
validateCertificate	<p><b>参数解释:</b> 是否验证服务端证书。</p> <p><b>取值范围:</b> true：验证服务端证书。 false：不验证服务端证书。</p> <p><b>默认取值:</b> false</p>	ObsConfiguration.setValidateCertificate	根据实际情况设置

参数	描述	方法	建议值
verifyResponseContentType	<p><b>参数解释:</b> 是否验证响应头信息的ContentType。</p> <p><b>取值范围:</b> true: 验证响应头信息的ContentType。 false: 不验证响应头信息的ContentType。</p> <p><b>默认取值:</b> true</p>	ObsConfiguration.setVerifyResponseType	默认
readBufferSize	<p><b>参数解释:</b> 从Socket流下载对象的缓存大小, -1表示不设置缓存。</p> <p><b>默认取值:</b> -1, 单位: 字节。</p>	ObsConfiguration.setReadBufferSize	根据实际情况设置
writeBufferSize	<p><b>参数解释:</b> 上传对象到Socket流时的缓存大小, -1表示不设置缓存。</p> <p><b>默认取值:</b> -1, 单位: 字节。</p>	ObsConfiguration.setWriteBufferSize	根据实际情况设置
socketWriteBufferSize	<p><b>参数解释:</b> Socket发送缓冲区大小, 对应java.net.SocketOptions.SO_SNDBUF参数。</p> <p><b>默认取值:</b> 默认为-1, 单位: 字节, 表示不设置。</p>	ObsConfiguration.setSocketWriteBufferSize	默认
socketReadBufferSize	<p><b>参数解释:</b> Socket接收缓冲区大小, 对应java.net.SocketOptions.SO_RCVBUF参数。</p> <p><b>默认取值:</b> 默认为-1, 单位: 字节, 表示不设置。</p>	ObsConfiguration.setSocketReadBufferSize	默认
keyManagerFactory	<p><b>参数解释:</b> 用于生成javax.net.ssl.KeyManager的工厂。</p> <p><b>默认取值:</b> 默认为空。</p>	ObsConfiguration.setKeyManagerFactory	根据实际情况设置

参数	描述	方法	建议值
trustManagerFactory	<p><b>参数解释:</b> 用于生成 javax.net.ssl.TrustManager 的工厂。</p> <p><b>默认取值:</b> 默认为空。</p>	ObsConfiguration.setTrustManagerFactory	根据实际情况设置
isStrictHostNameVerification	<p><b>参数解释:</b> 是否严格验证服务端主机名。设置为 true 后, 需要通过 javax.net.ssl.HttpsURLConnection.setDefaultHostnameVerifier 去设置一个实现 javax.net.ssl.HostnameVerifier 接口的对象, 用于验证主机名</p> <p><b>取值范围:</b> true: 严格验证服务端主机名。 false: 不严格验证服务端主机名。</p> <p><b>默认取值:</b> false</p>	ObsConfiguration.setIsStrictHostNameVerification	根据实际情况设置
keepAlive	<p><b>参数解释:</b> 是否使用长连接访问 OBS 服务。</p> <p><b>取值范围:</b> true: 使用长连接访问 OBS 服务。 false: 不使用长连接访问 OBS 服务。</p> <p><b>默认取值:</b> true</p>	ObsConfiguration.setKeepAlive	根据实际情况设置
cname	<p><b>参数解释:</b> 是否通过自定义域名访问 OBS 服务。</p> <p><b>取值范围:</b> true: 通过自定义域名访问 OBS 服务。 false: 不通过自定义域名访问 OBS 服务。</p> <p><b>默认取值:</b> false</p>	ObsConfiguration.setCname	根据实际情况设置

参数	描述	方法	建议值
sslProvider	<p><b>参数解释:</b> SSLContext的Provider。</p> <p><b>默认取值:</b> 使用JDK提供的SSLContext。</p>	ObsConfiguration.setSslProvider	根据实际情况设置
httpProtocolType	<p><b>参数解释:</b> 访问OBS服务端时使用的HTTP协议类型。</p> <p><b>默认取值:</b> HTTP1.1</p> <p><b>说明</b> 如果设置的endPoint不带协议类型，则默认使用HTTPS协议。</p>	ObsConfiguration.setHttpProtocolType	根据实际情况设置
httpDispatcher	<p><b>参数解释:</b> 自定义分发器。</p> <p><b>默认取值:</b> 无</p>	ObsConfiguration.setHttpDispatcher	根据实际情况设置
secureRandom	<p><b>参数解释:</b> 配置随机数生成器，支持设置为自定义的随机数生成器。</p> <p><b>默认取值:</b> 默认为 new SecureRandom()</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>在某些平台上new SecureRandom()可能会使用不安全实现。出于安全考虑，建议通过 ObsConfiguration.setSecureRandom()设置一个种子是真随机数的SecureRandom。</li> <li>当操作系统没有足够的熵来生成随机数时（比如系统刚启动时），那么生成真随机数的SecureRandom可能会阻塞，直到足够的熵可用为止。建议同时通过相关措施补充熵（例如linux下的haveged），改善熵源不足时生成真随机数阻塞的问题。</li> </ul>	ObsConfiguration.setSecureRandom	根据实际情况设置

## 6.3 配置 SDK 日志(Java SDK)

OBS Java SDK基于Apache Log4j2开源库提供了日志功能，SDK默认会将WARN级别的日志记录到JDK系统变量user.dir所代表的路径下，您可以通过修改日志配置文件定制日志功能。

## 操作步骤

**步骤1** 找到OBS Java SDK开发包中的log4j2.xml文件，或在[这里](#)获取。

**步骤2** 根据实际情况修改log4j2.xml中的日志级别和日志存放路径。

**步骤3** 将log4j2.xml文件放到classpath根目录，或者调用Log4j2Configurator.setLogConfig直接指定log4j2.xml文件的路径。

----**结束**

### 说明

- 您可以从[日志分析](#)章节获取更多关于SDK日志的信息。
- 您可以根据实际需要通过对修改log4j2.xml文件来配置日志文件权限。



# 7 管理桶(Java SDK)

## 7.1 创建桶(Java SDK)

### 功能说明

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Java 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以及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，如果初始化客户端使用的终端节点（endPoint）为“obs.myhuaweicloud.com”，则可以不指定桶所在区域（location），系统会自动在华北-北京一（cn-north-1）创建桶；如果初始化客户端使用的终端节点（endPoint）不是obs.myhuaweicloud.com，则必须指定桶所在区域（location），且指定的区域必须与终端节点（endPoint）区域一致，否则会返回状态码400。  
比如初始化时使用的终端节点endPoint是obs.cn-north-4.myhuaweicloud.com，那么在创建桶的时候必须指定Location: cn-north-4才会创建成功。
- 同一账号下，可以创建多个桶，数量上限是100个（包括对象桶和并行文件系统，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

- 方法1: 按照用户指定的桶名创建一个新桶  
createBucket(String bucketName)
- 方法2: 按照用户指定的桶名和指定的区域创建一个新桶  
createBucket(String bucketName, String location)
- 方法3: 按照用户指定的桶名和指定的区域创建一个新桶  
createBucket(ObsBucket bucket)
- 方法4: 创建桶, 支持设置访问权限、存储类别、区域位置、集群类型、桶类型 (对象桶或并行文件系统)、企业ID、桶扩展权限  
createBucket([CreateBucketRequest request](#))

## 请求参数说明

表 7-1 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">CreateBucketRequest</a>	是	<b>参数解释:</b> 创建桶请求参数, 详见 <a href="#">CreateBucketRequest</a> 。

表 7-2 CreateBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-.bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值：</b> 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><b>参数解释：</b> 桶所在的区域。</p> <p><b>约束限制：</b> 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><b>取值范围：</b> 当前有效的OBS区域和终端节点的更多信息，请参考<a href="#">地区和终端节点</a>。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><b>默认取值：</b> 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	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 创桶时可指定桶的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>
storageClasses	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 创桶时可指定的桶的存储类别。</p> <p><b>取值范围:</b> 可选择的访问策略选项参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> STANDARD，即标准存储类别。</p>
extensionPermissionMap	Map< <a href="#">ExtensionBucketPermissionEnum</a> , Set<String>>	可选	<p><b>参数解释:</b> 桶ACL的授权Map，您可以为一个或多个账号授予桶权限。Map的<a href="#">ExtensionBucketPermissionEnum</a>用于指定权限，Map的Set&lt;String&gt;用于说明该权限授予的账号ID列表，即domain_id列表。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 授予权限的范围详见<a href="#">ExtensionBucketPermissionEnum</a>。</li> <li>• 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a></li> </ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
epid	String	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 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><b>默认取值:</b> 无</p>
availableZone	<b>AvailableZoneEnum</b>	可选	<p><b>参数解释:</b> 创桶时可指定的数据冗余类型, 即AZ类型。</p> <p><b>约束限制:</b> 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><b>取值范围:</b> 如果将桶配置为多AZ, 则取值为"MULTI_AZ"。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><b>默认取值:</b> 不设置该参数, 则默认设置桶为单AZ。</p>
bucketType	<b>BucketTypeEnum</b>	可选	<p><b>参数解释:</b> 创建的桶类型。</p> <p><b>取值范围:</b> 详见<b>BucketTypeEnum</b>。</p> <p><b>默认取值:</b> OBJECT, 即对象桶。</p>

表 7-3 ACL 预定义访问策略

常量名	说明
AccessControllist.REST_CAN_NED_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	说明
AccessControlList.REST_CANNED_PUBLIC_READ	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。 设在对象上，所有人可以获取该对象内容和元数据。
AccessControlList.REST_CANNED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControlList.REST_CANNED_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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 7-4 ExtensionBucketPermissionEnum

常量名	描述
GRANT_READ	授予对应账号ID的账号READ权限。 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。
GRANT_WRITE	授予对应账号ID的账号WRITE权限。 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、拷贝段、合并段、取消多段上传任务。
GRANT_READ_ACP	授予对应账号ID的账号READ_ACP权限。 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。

常量名	描述
GRANT_WRITE_ACP	授予对应账号ID的账号WRITE_ACP权限。 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。
GRANT_FULL_CONTROL	授予对应账号ID的账号FULL_CONTROL权限。 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。
GRANT_READ_DELIVERED	授予对应账号ID的账号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。
GRANT_FULL_CONTROL_DELIVERED	授予对应账号ID的账号FULL_CONTROL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。

表 7-5 AccessControlList

参数名称	参数类型	是否必选	描述
owner	<a href="#">Owner</a>	可选	<b>参数解释：</b> 桶所有者的信息，详见 <a href="#">Owner</a> 。
delivered	boolean	可选	<b>参数解释：</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。 <b>取值范围：</b> true：是，桶ACL向桶内对象传递。 false：否，桶ACL不向桶内对象传递，仅作用于桶。 <b>默认取值：</b> false
grants	Set< <a href="#">GrantAndPermission</a> >	可选	<b>参数解释：</b> 被授权用户相关信息，详见 <a href="#">GrantAndPermission</a> 。

表 7-6 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 7-7 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<p><b>参数解释:</b> 被授权用户或用户组, 详见<a href="#">GranteeInterface</a>。</p>
permission	<a href="#">Permission</a>	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。</p> <p><b>默认取值:</b> false</p>



表 7-8 GranteeInterface

参数名称	参数类型	是否必选	描述
<b>CanonicalGrantee</b>	<b>CanonicalGrantee</b>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <b>CanonicalGrantee</b> 。
<b>GroupGrantee</b>	<b>GroupGrantee</b>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <b>GroupGrantee</b> 。 <b>默认取值:</b> 无

表 7-9 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 7-10 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 7-11 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。
PERMISSION_WRITE_ACP	WRITE_ACP	修改ACL配置的权限。 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。
PERMISSION_FULL_CONTROL	FULL_CONTROL	完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。 如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。 如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。

表 7-12 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。

常量名	原始值	说明
COLD	COLD	归档存储。

表 7-13 AvailableZoneEnum

常量名	原始值	说明
MULTI_AZ	3az	多az类型

表 7-14 BucketTypeEnum

常量名	原始值	说明
OBJECT	OBJECT	对象桶
PFS	POSIX	并行文件系统

## 返回结果说明

表 7-15 ObsBucket

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应头的名称，Object代表响应头的值。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
owner	<b>Owner</b>	<p><b>参数解释:</b> 桶拥有者信息，详见<b>Owner</b>。</p>
creationDate	java.util.Date	<p><b>参数解释:</b> 桶创建时间。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
location	String	<p><b>参数解释:</b> 桶所在的区域。</p> <p><b>约束限制:</b> 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><b>取值范围:</b> 当前有效的OBS区域和终端节点的更多信息，请参考<a href="#">地区和终端节点</a>。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><b>默认取值:</b> 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>
storageClass	<a href="#">StorageClassEnum</a>	<p><b>参数解释:</b> 创桶时可指定的桶的存储类别。</p> <p><b>取值范围:</b> 可选择的访问策略选项参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> STANDARD，即标准存储类别。</p>
acl	<a href="#">AccessControlList</a>	<p><b>参数解释:</b> 创桶时可指定桶的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"><li>● 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li><li>● 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li></ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>

参数名称	参数类型	描述
bucketTypeEnum	<b>BucketTypeEnum</b>	<p><b>参数解释:</b> 创建的桶类型。</p> <p><b>取值范围:</b> 详见<b>BucketTypeEnum</b>。</p> <p><b>默认取值:</b> OBJECT，即对象桶。</p>

## 代码示例：带参数创建

创建桶时可以指定桶的访问权限、存储类别和区域位置。示例代码如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
import com.obs.services.model.AvailableZoneEnum;
import com.obs.services.model.CreateBucketRequest;
import com.obs.services.model.ObsBucket;
import com.obs.services.model.StorageClassEnum;

public class CreateBucket001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");

        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";

        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            CreateBucketRequest request = new CreateBucketRequest();
            //示例桶名
            String exampleBucket = "examplebucket";
            //示例桶区域位置
            String exampleLocation = "cn-north-4";
            request.setBucketName(exampleBucket);
            // 设置桶访问权限为私有读写，默认也是私有读写
            request.setAcl(AccessControlList.REST_CANNED_PRIVATE);
            // 设置桶的存储类别为标准存储
            request.setBucketStorageClass(StorageClassEnum.STANDARD);
            // 设置桶区域位置(以区域为华北-北京四为例)，location 需要与 endpoint的位置信息一致
            request.setLocation(exampleLocation);
            // 指定创建多AZ桶，如果不设置，默认创建单AZ桶
            request.setAvailableZone(AvailableZoneEnum.MULTI_AZ);
            // 创建桶
            ObsBucket bucket = obsClient.createBucket(request);
        }
    }
}
```

```
// 创建桶成功
System.out.println("CreateBucket successfully");
System.out.println("RequestId:"+bucket.getRequestId());

} catch (ObsException e) {
    System.out.println("CreateBucket failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code: " + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message: " + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
} catch (Exception e) {
    System.out.println("CreateBucket failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：简单创建

以下代码展示如何新建一个桶，仅设置桶名和endPoint。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObsBucket;

public class CreateBucket002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint, 此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 示例桶名
            String exampleBucket = "examplebucket";
            // 创建桶
            ObsBucket bucket = obsClient.createBucket(exampleBucket);
            System.out.println("CreateBucket successfully");
            System.out.println("StatusCode: " + bucket.getStatusCode());
            System.out.println("RequestId: " + bucket.getRequestId());
        } catch (ObsException e) {
            System.out.println("CreateBucket failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code: " + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
        }
    }
}
```

```
// 请求失败,打印详细错误信息
System.out.println("Error Message: " + e.getErrorMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
} catch (Exception e) {
    System.out.println("CreateBucket failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于创建桶的API说明, 请参见[创建桶](#)。
- 更多关于创建桶的示例代码, 请参见[Github示例](#)。
- 创建桶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](#)。
- 创建桶常见问题请参见[创建桶失败](#)。

## 7.2 列举桶(Java SDK)

### 功能说明

OBS桶是对象的容器, 您上传的文件都存放在桶中。调用列举桶接口, 可列举当前账号所有地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以及region与endPoint的对应关系, 详细信息请参见[地区与终端节点](#)。
- 设置ListBucketsRequest.setQueryLocation参数为true后, 可在列举桶时查询桶的区域位置。

### 方法定义

obsClient.listBuckets([ListBucketsRequest request](#))

### 请求参数说明

表 7-16 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">ListBucketsRequest</a>	必选	<b>参数解释:</b> 获取桶列表请求参数, 详见 <a href="#">ListBucketsRequest</a> 。



表 7-17 ListBucketsRequest

参数名称	参数类型	是否必选	描述
queryLocation	boolean	可选	<p><b>参数解释:</b> 是否同时查询桶的区域位置。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• true: 同时查询桶的区域位置</li> <li>• false: 不查询桶的区域位置</li> </ul> <p><b>默认取值:</b> false</p>
bucketType	<a href="#">BucketTypeEnum</a>	可选	<p><b>参数解释:</b> 列举指定类型的桶。</p> <p><b>取值范围:</b> 详见<a href="#">BucketTypeEnum</a>。</p> <p><b>默认取值:</b> OBJECT, 即对象桶。</p>

表 7-18 BucketTypeEnum

常量名	原始值	说明
OBJECT	OBJECT	对象桶
PFS	POSIX	并行文件系统

## 返回结果说明

表 7-19 ListBucketsResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
owner	<b>Owner</b>	<p><b>参数解释:</b> 桶所有者信息，详见<b>Owner</b>。</p> <p><b>默认取值:</b> 无</p>
buckets	List< <b>ObsBucket</b> >	<p><b>参数解释:</b> 桶列表信息。详见<b>ObsBucket</b>。</p> <p><b>默认取值:</b> 无</p>

表 7-20 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID，即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a></p> <p><b>默认取值:</b> 无</p>
displayName	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a></p> <p><b>默认取值:</b> 无</p>

表 7-21 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。

常量名	原始值	说明
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 7-22 AccessControlList

参数名称	参数类型	是否必选	描述
owner	<a href="#">Owner</a>	可选	<b>参数解释:</b> 桶所有者的信息, 详见 <a href="#">Owner</a> 。
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false
grants	Set< <a href="#">GrantAndPermission</a> >	可选	<b>参数解释:</b> 被授权用户相关信息, 详见 <a href="#">GrantAndPermission</a> 。

表 7-23 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组, 详见 <a href="#">GranteeInterface</a> 。
permission	<a href="#">Permission</a>	必选	<b>参数解释:</b> 用户或用户组被授予的权限。 <b>取值范围:</b> 详见 <a href="#">Permission</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 7-24 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<p><b>参数解释:</b> 被授权用户的用户信息，详见<a href="#">CanonicalGrantee</a>。</p>
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<p><b>参数解释:</b> 被授权用户组的用户组信息。</p> <p><b>取值范围:</b> 详见<a href="#">GroupGrantee</a>。</p> <p><b>默认取值:</b> 无</p>

表 7-25 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<p><b>参数解释:</b> 被授权用户的账号ID，即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
displayName	String	可选	<p><b>参数描述:</b> 被授权用户的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 7-26 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户，已废弃。
LOG_DELIVERY	日志投递组，已废弃。

表 7-27 Permission

常量名	原始值	说明
PERMISSION_READ	READ	<p>读权限。</p> <p>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。</p> <p>如果有对象的读权限，则可以获取该对象内容和元数据。</p>
PERMISSION_WRITE	WRITE	<p>写权限。</p> <p>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。</p> <p>此权限在对象上不适用。</p>
PERMISSION_READ_ACP	READ_ACP	<p>读取ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有读对应桶或对象ACP的权限。</p>

常量名	原始值	说明
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

表 7-28 ACL 预定义访问策略

常量名	说明
AccessControlList.REST_CANNED_PRIVATE	<p>私有读写。</p> <p>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。</p>
AccessControlList.REST_CANNED_PUBLIC_READ	<p>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。</p> <p>设在对象上，所有人可以获取该对象内容和元数据。</p>
AccessControlList.REST_CANNED_PUBLIC_READ_WRITE	<p>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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务。</p> <p>设在对象上，所有人可以获取该对象内容和元数据。</p>

常量名	说明
AccessControlList.REST_CANNED_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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 7-29 ObsBucket

参数名称	参数类型	描述
statusCode	int	<b>参数解释：</b> HTTP状态码。 <b>取值范围：</b> 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 <a href="#">状态码</a> 。 <b>默认取值：</b> 无
responseHeaders	Map<String, Object>	<b>参数解释：</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值：</b> 无

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
owner	<b>Owner</b>	<p><b>参数解释:</b> 桶拥有者信息，详见<b>Owner</b>。</p>
creationDate	java.util.Date	<p><b>参数解释:</b> 桶创建时间。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	描述
location	String	<p><b>参数解释:</b> 桶所在的区域。</p> <p><b>约束限制:</b> 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><b>取值范围:</b> 当前有效的OBS区域和终端节点的更多信息，请参考<a href="#">地区和终端节点</a>。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><b>默认取值:</b> 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>
storageClass	<a href="#">StorageClassEnum</a>	<p><b>参数解释:</b> 创桶时可指定的桶的存储类别。</p> <p><b>取值范围:</b> 可选择的访问策略选项参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> STANDARD，即标准存储类别。</p>
acl	<a href="#">AccessControlList</a>	<p><b>参数解释:</b> 创桶时可指定桶的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"><li>● 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li><li>● 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li></ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>

参数名称	参数类型	描述
bucketTypeEnum	BucketTypeEnum	<p><b>参数解释:</b> 创建的桶类型。</p> <p><b>取值范围:</b> 详见<a href="#">BucketTypeEnum</a>。</p> <p><b>默认取值:</b> OBJECT，即对象桶。</p>

## 代码示例

本示例用于列举所有桶列表，并查询桶的区域。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListBucketsRequest;
import com.obs.services.model.ObsBucket;
import java.util.List;
public class ListBucket001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 列举桶
            ListBucketsRequest request = new ListBucketsRequest();
            request.setQueryLocation(true);
            List<ObsBucket> buckets = obsClient.listBuckets(request);
            for (ObsBucket bucket : buckets) {
                System.out.println("ListBuckets successfully");
                System.out.println("BucketName:" + bucket.getBucketName());
                System.out.println("CreationDate:" + bucket.getCreationDate());
                System.out.println("Location:" + bucket.getLocation());
            }
        } catch (ObsException e) {
            System.out.println("ListBuckets failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code: " + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
        }
    }
}
```

```
} catch (Exception e) {  
    System.out.println("ListBuckets failed");  
    // 其他异常信息打印  
    e.printStackTrace();  
}  
}
```

## 相关链接

- 关于列举桶的API说明，请参见[获取桶列表](#)。
- 获取桶列表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](#)。

## 7.3 删除桶(Java SDK)

### 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。
- 您必须是桶所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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 如果桶不为空（包含对象或分段上传碎片），则该桶无法被删除。
- 删除桶不是“幂等”操作，删除不存在的桶会报错。

### 方法定义

```
obsClient.deleteBucket(String bucketName)
```

## 请求参数说明

表 7-30 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>

## 返回结果

表 7-31 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<b>参数解释:</b> HTTP响应消息头列表, 由多个元组构成。元组中String代表响应消息头的名称, Object代表响应消息头的值。 <b>默认取值:</b> 无

## 代码示例

本示例用于删除桶名为exampleBucket的桶。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HeaderResponse;
public class DeleteBucket001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK, 也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以通过登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken, 也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint, 此处以华北-北京四为例, 其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint, 也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            //示例桶名
            String exampleBucket = "examplebucket";
            // 删除桶
            HeaderResponse response = obsClient.deleteBucket(exampleBucket);
            System.out.println("DeleteBucket successfully");
            System.out.println("StatusCode:"+response.getStatusCode());
            System.out.println("RequestId:"+response.getRequestId());
        } catch (ObsException e) {
            System.out.println("DeleteBucket failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message: " + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
        } catch (Exception e) {
            System.out.println("DeleteBucket failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

```
}  
}
```

## 相关链接

- 如何删除桶内对象和历史版本，请参见[删除对象](#)。
- 如何清理碎片，请参见[清理碎片](#)。
- 您可以使用[列举桶内对象](#)和[列举多段上传任务](#)来确认桶是否为空。
- 关于删除桶的API说明，请参见[删除桶](#)。
- 更多关于删除桶的示例代码，请参见[Github示例](#)。
- 删除桶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](#)。
- 删除桶常见问题请参见[删除桶失败](#)。

## 7.4 判断桶是否存在(Java SDK)

### 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以及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(String bucketName)
```

## 请求参数说明

表 7-32 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	是	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值：</b> 无</p>

## 返回结果说明

表 7-33 返回结果

参数类型	描述
boolean	<p><b>参数解释：</b> 桶是否存在。</p> <p><b>取值范围：</b> true：桶存在 false：桶不存在</p> <p><b>默认取值：</b> 无</p>

## 代码示例

本示例用于判断examplebucket桶是否存在。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class HeadBucket001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            //示例桶名
            String exampleBucket = "examplebucket";
            // 判断桶是否存在
            boolean exists = obsClient.headBucket(exampleBucket);
            System.out.println("HeadBucket successfully");
            System.out.println(exists);
        } catch (ObsException e) {
            System.out.println("HeadBucket failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code :"+ e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
        } catch (Exception e) {
            System.out.println("HeadBucket failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 判断桶是否存在和获取桶元数据是同一个REST API，关于判断桶是否存在的API说明，请参见[获取桶元数据](#)。
- 创建桶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](#)。



## 7.5 获取桶元数据(Java SDK)

### 功能说明

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、冗余策略等信息。

### 接口约束

- 您必须是桶所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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- BucketMetadataInfoResult.getAllowMethods等方法的值参见桶的[跨域资源共享（CORS）](#)配置。

### 方法定义

obsClient.getBucketMetadata([BucketMetadataInfoRequest request](#))

### 请求参数说明

表 7-34 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">BucketMetadataInfoRequest</a>	必选	<b>参数解释：</b> 获取桶元数据请求参数，详见 <a href="#">BucketMetadataInfoRequest</a> 。

表 7-35 BucketMetadataInfoRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
origin	String	可选	<p><b>参数解释:</b> 预请求指定的跨域请求origin（通常为域名）。</p> <p><b>约束限制:</b> 允许多条匹配规则，以回车换行为间隔。每个匹配规则允许使用最多一个“*”通配符。</p> <p><b>默认取值:</b> 无</p>
requestHeaders	List<String>	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 允许的头域可设置多个，多个头域之间换行隔开，每行最多可填写一个*符号，不支持&amp;、:、&lt;、空格以及中文字符。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 7-36 BucketMetadataInfoResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。<b>默认取值:</b> 无</p>
location	String	<p><b>参数解释:</b> 桶所在的区域。</p> <p><b>约束限制:</b> 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><b>取值范围:</b> 当前有效的OBS区域和终端节点的更多信息，请参考<a href="#">地区和终端节点</a>。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><b>默认取值:</b> 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>
obsVersion	String	<p><b>参数解释:</b> 桶所在的OBS服务版本号。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"><li>“3.0”表示最新版本的桶。</li><li>“--”表示老版本的桶。</li></ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
storageClass	<a href="#">StorageClassEnum</a>	<b>参数解释:</b> 创桶时可指定的桶的存储类别。 <b>取值范围:</b> 可选择的访问策略选项参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> STANDARD, 即标准存储类别。
allowOrigin	String	<b>参数解释:</b> 如果请求中的Origin满足桶的CORS规则, 则返回CORS规则中的AllowedOrigin。 <b>约束限制:</b> 允许多条匹配规则, 以回车换行为间隔。每个匹配规则允许使用最多一个“*”通配符。 <b>默认取值:</b> 无
allowHeaders	List<String>	<b>参数解释:</b> 如果请求中的RequestHeader满足桶的CORS规则, 则返回CORS规则中的AllowedHeader。 <b>约束限制:</b> 允许的头域可设置多个, 多个头域之间换行隔开, 每行最多可填写一个*符号, 不支持&、:、<、空格以及中文字符。 <b>默认取值:</b> 无
allowMethods	List<String>	<b>参数解释:</b> 指定允许的跨域请求HTTP方法, 即桶和对象的几种操作类型。 <b>取值范围:</b> 支持以下HTTP方法: <ul style="list-style-type: none"><li>• GET</li><li>• PUT</li><li>• HEAD</li><li>• POST</li><li>• DELETE</li></ul> <b>默认取值:</b> 无

参数名称	参数类型	描述
exposeHeaders	List<String>	<b>参数解释:</b> 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。 <b>约束限制:</b> 不支持*、&、:、<、空格以及中文字符。 <b>默认取值:</b> 无
maxAge	int	<b>参数解释:</b> 请求来源的客户端可以对跨域请求返回结果的缓存时间。 <b>约束限制:</b> 每个CORSRule可以包含至多一个maxAge。 <b>取值范围:</b> 大于0的整型数，单位：秒。 <b>默认取值:</b> 100
epid	String	<b>参数解释:</b> 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。 <b>约束限制:</b> Epid格式为uuid，默认项目传“0”或者不带该头域，未开通企业项目的用户可以不带该头域。 示例：9892d768-2d13-450f-aac7-ed0e44c2585f <b>取值范围:</b> 获取方式参见 <a href="#">如何获取企业项目ID</a> 。 <b>默认取值:</b> 无
bucketType	<b>BucketType Enum</b>	<b>参数解释:</b> 创建的桶类型。 <b>取值范围:</b> 详见 <a href="#">BucketTypeEnum</a> 。 <b>默认取值:</b> OBJECT，即对象桶。

参数名称	参数类型	描述
availableZone	<a href="#">AvailableZoneEnum</a>	<p><b>参数解释:</b> 桶的数据冗余存储策略属性，反映数据是单AZ存储还是多AZ存储。详见<a href="#">AvailableZoneEnum</a></p> <p><b>约束限制:</b> 取值为3az，表示数据冗余存储在同一区域的多个可用区。 不携带此头域表示为单az存储，仅使用1个可用区存储。</p> <p><b>默认取值:</b> 无</p>

表 7-37 BucketTypeEnum

常量名	原始值	说明
OBJECT	OBJECT	对象桶
PFS	POSIX	并行文件系统

表 7-38 AvailableZoneEnum

常量名	原始值	说明
MULTI_AZ	3az	多az类型

表 7-39 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

## 代码示例

本示例用于获取examplebucket桶的元数据信息，指定的跨域请求origin为“http://www.exampleorigin.com”。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketMetadataInfoRequest;
import com.obs.services.model.BucketMetadataInfoResult;
public class GetBucketMetadata001 {
```

```
public static void main(String[] args) {
    // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
    // 您可以登录访问管理控制台获取访问密钥AK/SK
    String ak = System.getenv("ACCESS_KEY_ID");
    String sk = System.getenv("SECRET_ACCESS_KEY_ID");
    // 【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/SecurityToken，也可以使用其他外部引入方式传入。
    // String securityToken = System.getenv("SECURITY_TOKEN");
    // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
    String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
    // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
    // String endPoint = System.getenv("ENDPOINT");

    // 创建ObsClient实例
    // 使用永久AK/SK初始化客户端
    ObsClient obsClient = new ObsClient(ak, sk, endPoint);
    // 使用临时AK/SK和SecurityToken初始化客户端
    // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

    try {
        // 示例桶名
        String exampleBucket = "examplebucket";
        // 示例Origin
        String exampleOrigin = "http://www.exampleorigin.com";
        BucketMetadataInfoRequest request = new BucketMetadataInfoRequest(exampleBucket);
        // 只有在涉及 cors 时才需要 setOrigin
        request.setOrigin(exampleOrigin);
        // 获取桶元数据
        BucketMetadataInfoResult result = obsClient.getBucketMetadata(request);
        System.out.println("GetBucketMetadata successfully");
        System.out.println("GetBucketType:" + result.getBucketType());
        System.out.println("GetLocation:" + result.getLocation());
        System.out.println("GetBucketStorageClass:" + result.getBucketStorageClass());
        System.out.println("GetObsVersion:" + result.getObsVersion());
        System.out.println("GetAllowOrigin:" + result.getAllowOrigin());
        System.out.println("GetMaxAge:" + result.getMaxAge());
        System.out.println("GetAllowHeaders:" + result.getAllowHeaders());
        System.out.println("GetAllowMethods:" + result.getAllowMethods());
        System.out.println("GetExposeHeaders:" + result.getExposeHeaders());
    } catch (ObsException e) {
        System.out.println("GetBucketMetadata failed");
        // 请求失败,打印http状态码
        System.out.println("HTTP Code:" + e.getResponseCode());
        // 请求失败,打印服务端错误码
        System.out.println("Error Code:" + e.getErrorCode());
        // 请求失败,打印详细错误信息
        System.out.println("Error Message:" + e.getErrorMessage());
        // 请求失败,打印请求id
        System.out.println("Request ID:" + e.getErrorRequestId());
        System.out.println("Host ID:" + e.getErrorHostId());
        e.printStackTrace();
    } catch (Exception e) {
        System.out.println("GetBucketMetadata failed");
        // 其他异常信息打印
        e.printStackTrace();
    }
}
```

## 相关链接

- 关于获取桶元数据的API说明，请参见[获取桶元数据](#)。
- 更多关于获取桶元数据的示例代码，请参见[Github示例](#)。
- 获取桶元数据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](#)。

## 7.6 设置桶 ACL(Java SDK)

### 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](#)。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

桶访问权限（[ACL](#)）可以通过三种方式设置，具体参见[方法定义](#)。

OBS支持的桶或对象权限包含五类，见[表Permission](#)。

OBS预定义的访问策略包含五类，见下表[ACL预定义访问策略](#)。

### 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以及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

- 方法1：创建桶时，设置访问权限：

```
obsBucket.setBucketName(exampleBucket);  
// 设置桶访问权限为私有读写  
obsBucket.setAcl(AccessControlList.REST_CANNED_PRIVATE);  
// 创建桶  
obsClient.createBucket(obsBucket);
```
- 方法2：通过预定义访问策略设置访问权限：

```
// 设置桶访问权限为私有读写  
obsClient.setBucketAcl(String exampleBucket, AccessControlList.REST_CANNED_PRIVATE);
```
- 方法3：调用ObsClient.setBucketAcl直接设置访问权限：



```
// 直接设置桶访问权限
obsClient.setBucketAcl(String bucketName,AccessControllist acl);
```

## 请求参数说明

表 7-40 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
acl	<a href="#">AccessControllist</a>	必选	<p><b>参数解释:</b> 创桶时可指定桶的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>如果选择自定义ACL策略，则可以根据<a href="#">AccessControllist</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> AccessControllist.REST_CANNED_PRIVATE</p>

表 7-41 AccessControllist

参数名称	参数类型	是否必选	参数类型
owner	<b>Owner</b>	必选	<b>参数解释:</b> 桶所有者的信息, 详见 <b>Owner</b> 。
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false
grants	Set< <b>GrantAndPermission</b> >	可选	<b>参数解释:</b> 被授权用户相关信息, 详见 <b>GrantAndPermission</b> 。

表 7-42 预定义的 AccessControllist

常量名	描述
AccessControllist.REST_CANNED_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。
AccessControllist.REST_CANNED_PUBLIC_READ	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。 设在对象上, 所有人可以获取该对象内容和元数据。
AccessControllist.REST_CANNED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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	描述
AccessControlList.REST_CANNED_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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 7-43 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释：</b> 桶所有者的账号ID，即domain_id。 <b>取值范围：</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值：</b> 无
display Name	String	可选	<b>参数解释：</b> 所有者的账号名。 <b>取值范围：</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值：</b> 无

表 7-44 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释：</b> 被授权用户或用户组，详见 <a href="#">GranteeInterface</a> 。

参数名称	参数类型	是否必选	描述
permission	<a href="#">Permission</a>	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 7-45 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<p><b>参数解释:</b> 被授权用户的用户信息，详见<a href="#">CanonicalGrantee</a>。</p>
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<p><b>参数解释:</b> 被授权用户组的用户组信息。</p> <p><b>取值范围:</b> 详见<a href="#">GroupGrantee</a>。</p> <p><b>默认取值:</b> 无</p>

表 7-46 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<p><b>参数解释:</b> 被授权用户的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
displayName	String	可选	<p><b>参数描述:</b> 被授权用户的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 7-47 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 7-48 Permission

常量名	原始值	说明
PERMISSION_READ	READ	<p>读权限。</p> <p>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。</p> <p>如果有对象的读权限, 则可以获取该对象内容和元数据。</p>
PERMISSION_WRITE	WRITE	<p>写权限。</p> <p>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。</p> <p>此权限在对象上不适用。</p>

常量名	原始值	说明
PERMISSION_READ_ACP	READ_ACP	<p>读取ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有读对应桶或对象ACP的权限。</p>
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

## 返回结果说明

表 7-49 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释：</b> HTTP状态码。</p> <p><b>取值范围：</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值：</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例：创建桶时指定预定义访问策略

本示例用于创建exampleBucket桶时指定预定义访问策略。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
import com.obs.services.model.ObsBucket;
public class SetBucketAcl001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            ObsBucket obsBucket = new ObsBucket();
            //示例桶名
            String exampleBucket = "examplebucket";
            obsBucket.setBucketName(exampleBucket);
            // 设置桶访问权限为私有读写
            obsBucket.setAcl(AccessControlList.REST_CANNED_PRIVATE);
            // 创建桶
            obsClient.createBucket(obsBucket);
            System.out.println("SetBucketAcl successfully");
        } catch (ObsException e) {
            System.out.println("SetBucketAcl failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("SetBucketAcl failed");
        }
    }
}
```

```
        // 其他异常信息打印
        e.printStackTrace();
    }
}
}
```

## 代码示例：为桶设置预定义访问策略

本示例用于为exampleBucket桶设置预定义访问策略。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
public class SetBucketAcl002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            //示例桶名
            String exampleBucket = "examplebucket";
            // 设置桶访问权限为私有读写
            obsClient.setBucketAcl(exampleBucket, AccessControlList.REST_CANNED_PRIVATE);
            System.out.println("SetBucketAcl successfully");
        } catch (ObsException e) {
            System.out.println("SetBucketAcl failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("SetBucketAcl failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 代码示例：设置桶的访问策略

本示例用于直接设置exampleBucket桶的访问策略。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
```



```
import com.obs.services.model.CanonicalGrantee;
import com.obs.services.model.GroupGrantee;
import com.obs.services.model.Owner;
import com.obs.services.model.Permission;
public class SetBucketAcl003 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            //示例桶名
            String exampleBucket = "examplebucket";
            //示例userid
            String exampleUserId = "userid";
            //示例ownerid
            String exampleOwnerId = "ownerid";
            AccessControlList acl = new AccessControlList();
            Owner owner = new Owner();
            owner.setId(exampleOwnerId);
            acl.setOwner(owner);
            // 为指定用户设置完全控制权限
            acl.grantPermission(new CanonicalGrantee(exampleUserId),
                Permission.PERMISSION_FULL_CONTROL);
            // 为所有用户设置读权限
            acl.grantPermission(GroupGrantee.ALL_USERS, Permission.PERMISSION_READ);
            // 直接设置桶访问权限
            obsClient.setBucketAcl(exampleBucket, acl);
            System.out.println("SetBucketAcl successfully");
        } catch (ObsException e) {
            System.out.println("SetBucketAcl failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("SetBucketAcl failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于设置桶ACL的API说明，请参见[设置桶ACL](#)。

- 更多关于桶ACL的示例代码，请参见[Github示例](#)。
- 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](#)。

## 7.7 获取桶 ACL(Java 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

- 您必须是桶拥有者或拥有获取桶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以及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(String bucketName)
```

## 获取桶访问权限请求参数说明

表 7-50 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 7-51 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	<b>Owner</b>	必选	<p><b>参数解释:</b> 桶所有者的信息，详见<b>Owner</b>。</p>

参数名称	参数类型	是否必选	参数类型
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。 <b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。 <b>默认取值:</b> false
grants	Set< <a href="#">GrantAndPermission</a> >	可选	<b>参数解释:</b> 被授权用户相关信息，详见 <a href="#">GrantAndPermission</a> 。

表 7-52 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID，即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> <b>默认取值:</b> 无
displayName	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> <b>默认取值:</b> 无

表 7-53 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组，详见 <a href="#">GranteeInterface</a> 。

参数名称	参数类型	是否必选	描述
permission	Permission	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见Permission。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 7-54 GranteeInterface

参数名称	参数类型	是否必选	描述
CanonicalGrantee	CanonicalGrantee	必选	<p><b>参数解释:</b> 被授权用户的用户信息，详见CanonicalGrantee。</p>
GroupGrantee	GroupGrantee	必选	<p><b>参数解释:</b> 被授权用户组的用户组信息。</p> <p><b>取值范围:</b> 详见GroupGrantee。</p> <p><b>默认取值:</b> 无</p>

表 7-55 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<p><b>参数解释:</b> 被授权用户的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
displayName	String	可选	<p><b>参数描述:</b> 被授权用户的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 7-56 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 7-57 Permission

常量名	原始值	说明
PERMISSION_READ	READ	<p>读权限。</p> <p>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。</p> <p>如果有对象的读权限, 则可以获取该对象内容和元数据。</p>
PERMISSION_WRITE	WRITE	<p>写权限。</p> <p>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。</p> <p>此权限在对象上不适用。</p>

常量名	原始值	说明
PERMISSION_READ_ACP	READ_ACP	<p>读取ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有读对应桶或对象ACP的权限。</p>
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

## 代码示例

本示例用于获取examplebucket桶中objectname对象的访问权限。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
public class GetBucketAcl001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
```

```
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 示例桶名
    String exampleBucket = "examplebucket";
    // 获取桶访问权限
    AccessControlList acl = obsClient.getBucketAcl(exampleBucket);
    System.out.println("GetBucketAcl successfully");
    System.out.println(acl);
} catch (ObsException e) {
    System.out.println("GetBucketAcl failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("GetBucketAcl failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于获取桶ACL的API说明, 请参见[获取桶ACL](#)。
- 更多关于桶ACL的示例代码, 请参见[Github示例](#)。
- 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](#)。

## 7.8 设置桶策略(Java SDK)

### 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, 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。允许其他云账号访问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。

#### 须知

开发过程中, 您有任何问题可以在github上[提交issue](#), 或者在[华为云对象存储服务论坛](#)中发帖求助。

除了桶访问权限外, 桶的拥有者还可以通过桶策略, 提供对桶和桶内对象的集中访问控制。



您可以通过ObsClient.setBucketPolicy设置桶策略。具体格式请参考[Policy格式](#)，桶策略说明请参考[OBS权限控制概述](#)，桶策略样例请参考[桶策略样例](#)。

更多关于桶策略的内容请参考[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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 请注意，调用设置桶策略接口时，最新一次配置会覆盖原有配置。例如，您已设置了A，B，C和D四个桶策略，现需要添加新的桶策略E，则配置桶策略E时，需在原有四个策略的基础上添加策略E，然后重新上传所有策略。同理，如果您需要删除桶策略D，则需将策略D从原有的A，B，C和D四个策略中移除，然后重新上传策略A，B和C。
- 桶策略内容的具体格式（JSON格式字符串）请参考[对象存储服务API参考](#)。
- 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。

## 方法定义

```
obsClient.setBucketPolicy(String bucketName, String policy)
```

## 设置桶策略请求参数说明

表 7-58 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值：</b> 无</p>
policy	String	必选	<p><b>参数解释：</b> 策略信息，JSON格式的字符串。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"><li>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。</li><li>Policy具体格式请参考<a href="#">Policy格式</a>。</li></ul> <p><b>默认取值：</b> 无</p>

## 返回结果说明

表 7-59 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于设置exampleBucket桶的桶策略。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class SetBucketPolicy001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 示例桶名
            String exampleBucket = "examplebucket";
            // 示例桶策略
            String examplePolicy =
```

```
        {"Statement":[{"Principal":{"*"},"Effect":"Allow","Action":["ListBucket"],"Resource\n\n\n                + exampleBucket\n                + "\\}"}];\n        obsClient.setBucketPolicy(exampleBucket, examplePolicy);\n        System.out.println("SetBucketAcl successfully");\n    } catch (ObsException e) {\n        System.out.println("SetBucketAcl failed");\n        // 请求失败,打印http状态码\n        System.out.println("HTTP Code:" + e.getResponseCode());\n        // 请求失败,打印服务端错误码\n        System.out.println("Error Code:" + e.getErrorCode());\n        // 请求失败,打印详细错误信息\n        System.out.println("Error Message:" + e.getErrorMessage());\n        // 请求失败,打印请求id\n        System.out.println("Request ID:" + e.getErrorRequestId());\n        System.out.println("Host ID:" + e.getErrorHostId());\n        e.printStackTrace();\n    } catch (Exception e) {\n        System.out.println("SetBucketAcl failed");\n        // 其他异常信息打印\n        e.printStackTrace();\n    }\n}\n}
```

## 相关链接

- 关于设置桶策略的API说明，请参见[设置桶策略](#)。
- 更多关于桶策略的示例代码，请参见[Github示例](#)。
- 设置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](#)。

## 7.9 获取桶策略(Java 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。

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以及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(String **bucketName**)

## 获取桶策略请求参数说明

表 7-60 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 7-61 返回结果列表

参数名称	参数类型	描述
policy	String	<p><b>参数解释:</b> 策略信息，JSON格式的字符串。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>Policy具体格式请参考<a href="#">Policy格式</a>。</li> </ul> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于获取exampleBucket桶的桶策略。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class GetBucketPolicy001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 示例桶名
            String exampleBucket = "examplebucket";
            String policy = obsClient.getBucketPolicy(exampleBucket);
            System.out.println("\t" + policy);
        } catch (ObsException e) {
            System.out.println("GetBucketAcl failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        }
    }
}
```

```
} catch (Exception e) {  
    System.out.println("GetBucketAcl failed");  
    // 其他异常信息打印  
    e.printStackTrace();  
}  
}
```

## 相关链接

- 关于获取桶策略的API说明，请参见[获取桶策略](#)。
- 更多关于桶策略的示例代码，请参见[Github示例](#)。
- 设置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](#)。

## 7.10 删除桶策略(Java 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。

调用ObsClient.deleteBucketPolicy接口，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以及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(String [bucketName](#))

## 删除桶策略请求参数说明

表 7-62 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值：</b> 无</p>

## 返回结果说明

表 7-63 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释：</b> HTTP状态码。</p> <p><b>取值范围：</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值：</b> 无</p>



参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<b>参数解释:</b> HTTP响应消息头列表, 由多个元组构成。元组中String代表响应消息头的名称, Object代表响应消息头的值。 <b>默认取值:</b> 无

## 代码示例

本示例用于删除exampleBucket桶的桶策略。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class DeleteBucketPolicy001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK, 也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以通过登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken, 也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint, 此处以华北-北京四为例, 其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint, 也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 示例桶名
            String exampleBucket = "examplebucket";
            obsClient.deleteBucketPolicy(exampleBucket);
        } catch (ObsException e) {
            System.out.println("DeleteBucketAcl failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("DeleteBucketAcl failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

关于删除桶策略的API说明，请参见[删除桶策略](#)。

更多关于桶策略的示例代码，请参见[Github示例](#)。

设置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](#)。

## 7.11 获取桶区域位置(Java SDK)

### 功能说明

您可以通过ObsClient.getBucketLocation获取桶的区域位置。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 创建桶时可以指定桶的区域位置，请参见[创建桶](#)。

### 方法定义

obsClient.getBucketLocation(String [bucketName](#))

## 请求参数说明

表 7-64 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 7-65 BucketLocationResponse

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
location	String	<p><b>参数解释:</b> 桶所在的区域。</p> <p><b>约束限制:</b> 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><b>取值范围:</b> 当前有效的OBS区域和终端节点的更多信息，请参考<a href="#">地区和终端节点</a>。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><b>默认取值:</b> 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>

## 代码示例

本示例用于获取exampleBucket桶的桶区域位置。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class GetBucketLocation001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
```

```
//示例桶名
String exampleBucket = "examplebucket";
//获取桶区域位置
String location = obsClient.getBucketLocation(exampleBucket);
System.out.println("GetBucketLocation successfully");
System.out.println("Location:" + location);
} catch (ObsException e) {
    System.out.println("GetBucketLocation failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("GetBucketLocation failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于获取桶区域位置的API说明，请参见[获取桶区域位置](#)。
- 更多关于获取桶区域位置的示例代码，请参见[Github示例](#)。
- 获取桶区域位置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](#)。

## 7.12 获取桶存量信息(Java SDK)

### 功能说明

桶存量信息包括桶已使用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。

#### 说明

由于OBS桶存量是后台统计，因此存量会有一定的时延，不能实时更新，因此不建议对存量做实时校验。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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:GetBucketStorageInfo` 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以及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(String bucketName)
```

## 请求参数说明

表 7-66 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"><li>• 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。</li><li>• 桶命名规则如下：<ul style="list-style-type: none"><li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>- 禁止使用IP地址。</li><li>- 禁止以“-”或“.”开头及结尾。</li><li>- 禁止两个“.”相邻（如：“my..bucket”）。</li><li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>• 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。</li></ul> <p><b>默认取值：</b> 无</p>

## 返回结果说明

表 7-67 BucketStorageInfo

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
size	long	<b>参数解释:</b> 桶中对象占用的空间大小。 <b>取值范围:</b> 值为非负整数，单位为Byte（字节）。 <b>默认取值:</b> 无
objectNum	long	<b>参数解释:</b> 桶内对象个数。 <b>默认取值:</b> 无

## 代码示例

本示例用于获取exampleBucket桶的桶存量信息。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketStorageInfo;
public class GetBucketStorageInfo001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
```

```
// endpoint填写桶所在的endpoint, 此处以华北-北京四为例, 其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint, 也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    //示例桶名
    String exampleBucket = "examplebucket";
    //获取桶存量信息
    BucketStorageInfo storageInfo = obsClient.getBucketStorageInfo(exampleBucket);
    System.out.println("GetBucketStorageInfo successfully");
    System.out.println("GetObjectNumber:" + storageInfo.getObjectNumber());
    System.out.println("GetStorageSize:" + storageInfo.getSize() + " bytes");
} catch (ObsException e) {
    System.out.println("GetBucketStorageInfo failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("GetBucketStorageInfo failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于获取桶存量信息的API说明, 请参见[获取桶存量信息](#)。
- 更多关于获取桶存量信息的示例代码, 请参见[Github示例](#)。
- 获取桶存量信息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](#)。

## 7.13 设置桶配额(Java SDK)

### 功能说明

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。



### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 接口约束

- 桶配额值必须为非负整数，单位为字节。
- 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以及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(String bucketName, BucketQuota bucketQuota)
```

## 设置桶配额请求参数说明

表 7-68 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
bucketQuota	<b>BucketQuota</b>	必选	<p><b>参数解释:</b> 桶的配额信息</p> <p><b>约束限制:</b> 详见<a href="#">BucketQuota</a></p>

表 7-69 BucketQuota

参数名称	参数类型	是否必选	描述
bucketQuota	long	必选	<p><b>参数解释:</b> 桶的配额值。</p> <p><b>取值范围:</b> 非负整数，单位：字节。</p> <p><b>默认取值:</b> 0，表示没有限制桶配额。</p>

## 返回结果说明

表 7-70 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于设置exampleBucket桶的桶配额信息。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketQuota;
public class SetBucketQuota001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 示例桶名
            String exampleBucket = "examplebucket";
            // 示例桶配额
```

```
long exampleBucketQuota = 1024 * 1024 * 100L;
// 设置桶配额为100MB
BucketQuota quota = new BucketQuota(exampleBucketQuota);
obsClient.setBucketQuota(exampleBucket, quota);
System.out.println("SetBucketQuota successfully");
} catch (ObsException e) {
    System.out.println("SetBucketQuota failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("SetBucketQuota failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置桶配额的API说明，请参见[设置桶配额](#)。
- 更多关于桶配额的示例代码，请参见[Github示例](#)。
- 设置桶配额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](#)。

## 7.14 获取桶配额(Java 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。
- 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以及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(String bucketName)
```

## 获取桶配额请求参数说明

表 7-71 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“-”和“.”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值：</b> 无</p>

## 返回结果说明

表 7-72 BucketQuota

参数名称	参数类型	是否必选	描述
bucketQuota	long	必选	<p><b>参数解释：</b> 桶的配额值。</p> <p><b>取值范围：</b> 非负整数，单位：字节。</p> <p><b>默认取值：</b> 0，表示没有限制桶配额。</p>

## 代码示例

本示例用于获取exampleBucket桶的桶配额信息。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketQuota;
public class GetBucketQuota001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以通过登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            //示例桶名
            String exampleBucket = "examplebucket";
            //获取桶配额
            BucketQuota quota = obsClient.getBucketQuota(exampleBucket);
            System.out.println("GetBucketQuota successfully");
            System.out.println("GetBucketQuota:" + quota.getBucketQuota() + " bytes");
        } catch (ObsException e) {
            System.out.println("GetBucketQuota failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("GetBucketQuota failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于获取桶配额的API说明，请参见[获取桶配额](#)。
- 更多关于桶配额的示例代码，请参见[Github示例](#)。
- 设置桶配额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](#)。

## 7.15 设置桶存储类别(Java SDK)

### 功能说明

OBS提供了四种存储类别：标准存储、低频访问存储、归档存储、深度归档存储（受限公测中），从而满足客户业务对存储性能、成本的不同诉求，详情可参见OBS存储类别存储类别存储类别。

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，如果上传对象、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。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

桶的存储类别分为三类，见下表：

类型	说明	OBS Java SDK对应值
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。	StorageClassEnum.STANDARD
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。	StorageClassEnum.WARM
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。	StorageClassEnum.COLD

更多关于桶存储类别的内容请参考[桶的存储类别](#)。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 桶的存储类别与桶中对象的存储类别是相互独立的。如果上传对象时，未设置对象的存储类别，该对象默认与桶的存储类别一致，但后续如果修改桶的存储类别，已上传的桶中对象的存储类别不会随之改变；如果后续修改桶中对象的存储类别时，该桶的存储类别也不会随之改变。

## 方法定义

```
obsClient.setBucketStoragePolicy(String bucketName,  
BucketStoragePolicyConfiguration bucketStorage)
```

## 请求参数说明

表 7-73 参数列表说明

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
bucketStorage	<b>BucketStoragePolicyConfiguration</b> (继承HeaderResponse)	必选	<p><b>参数解释:</b> 桶的存储类别。</p> <p><b>取值范围:</b> 存储类别取值详见 <a href="#">BucketStoragePolicyConfiguration</a>。</p> <p><b>默认取值:</b> 无</p>



表 7-74 BucketStoragePolicyConfiguration

参数名称	参数类型	描述
storageClass	<a href="#">StorageClassEnum</a>	<b>参数解释:</b> 桶的存储类别。 <b>取值范围:</b> 存储类别取值详见桶的 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无

表 7-75 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

## 返回结果说明

表 7-76 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无

## 代码示例

本示例用于设置exampleBucket桶的桶存储类别。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
import com.obs.services.model.AvailableZoneEnum;
import com.obs.services.model.CreateBucketRequest;
import com.obs.services.model.ObsBucket;
import com.obs.services.model.StorageClassEnum;

public class CreateBucket001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");

        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";

        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            CreateBucketRequest request = new CreateBucketRequest();
            //示例桶名
            String exampleBucket = "examplebucket";
            //示例桶区域位置
            String exampleLocation = "cn-north-4";
            request.setBucketName(exampleBucket);
            // 设置桶访问权限为私有读写，默认也是私有读写
            request.setAcl(AccessControlList.REST_CANNED_PRIVATE);
            // 设置桶的存储类别为标准存储
            request.setBucketStorageClass(StorageClassEnum.STANDARD);
            // 设置桶区域位置(以区域为华北-北京四为例)，location 需要与 endpoint的位置信息一致
            request.setLocation(exampleLocation);
            // 指定创建多AZ桶，如果不设置，默认创建单AZ桶
            request.setAvailableZone(AvailableZoneEnum.MULTI_AZ);
            // 创建桶
            ObsBucket bucket = obsClient.createBucket(request);
            // 创建桶成功
            System.out.println("CreateBucket successfully");
            System.out.println("RequestId:" + bucket.getRequestId());

        } catch (ObsException e) {
            System.out.println("CreateBucket failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code: " + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message: " + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
        }
    }
}
```

```
} catch (Exception e) {  
    System.out.println("CreateBucket failed");  
    // 其他异常信息打印  
    e.printStackTrace();  
}  
}
```

## 相关链接

- 关于设置桶存储类别的API说明，请参见[设置桶存储类别](#)。
- 更多关于桶存储类别的示例代码，请参见[Github示例](#)。
- 设置桶存储类别过程中返回的的错误码含义、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](#)。

## 7.16 获取桶存储类别(Java SDK)

### 功能介绍

OBS提供了四种存储类别：标准存储、低频访问存储、归档存储、深度归档存储（受限公测中），从而满足客户业务对存储性能、成本的不同诉求，详情可参见[OBS存储类别](#)。

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以及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(String [bucketName](#))

## 请求参数说明

表 7-77 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值：</b> 无</p>

## 返回结果说明

表 7-78 BucketStoragePolicyConfiguration

参数名称	参数类型	描述
storageClass	<a href="#">StorageClassEnum</a>	<p><b>参数解释：</b> 桶的存储类别，详见<a href="#">StorageClassEnum</a>。</p>

表 7-79 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。

常量名	原始值	说明
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

## 代码示例

本示例用于获取exampleBucket桶的桶存储类别。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketStoragePolicyConfiguration;
public class GetBucketStoragePolicy001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            //示例桶名
            String exampleBucket = "examplebucket";
            //获取桶存储类别
            BucketStoragePolicyConfiguration storagePolicy =
obsClient.getBucketStoragePolicy(exampleBucket);
            System.out.println("GetBucketStoragePolicy successfully");
            System.out.println("GetBucketStorageClass:" + storagePolicy.getBucketStorageClass());
        } catch (ObsException e) {
            System.out.println("GetBucketStoragePolicy failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("GetBucketStoragePolicy failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于获取桶存储类别的API说明，请参见[获取桶存储类别](#)。
- 更多关于桶存储类别的示例代码，请参见[Github示例](#)。
- 设置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](#)。

## 7.17 桶清单(Java SDK)

### 7.17.1 设置桶清单规则(Java SDK)

#### 功能说明

桶清单功能可以定期生成桶内对象的相关信息，保存在CSV格式的文件中，并上传到您指定的桶中，方便您管理桶内对象。目标桶和源桶可以是同一个桶。指定的对象内容包括对象版本、大小、上次修改时间、存储类别、标签、加密状态等。

支持对桶清单加密，加密方式为SSE-KMS方式。

#### 接口约束

- 您必须是桶拥有者或者拥有设置桶清单权限，才能设置桶清单。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:PutBucketInventoryConfiguration权限，如果使用桶策略则需授予PutBucketInventoryConfiguration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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- **桶版本限制：**
  - 只有OBS 3.0的桶支持配置清单，目标桶无OBS版本限制。
- **清单数量限制：**
  - 一个桶最多支持10条桶清单。
- **源桶和目标桶限制：**
  - 桶清单配置的源桶和目标桶必须归属同一个账号。
  - 桶清单配置的源桶和目标桶必须归属同一个区域。
  - 桶清单中配置的目标桶不能开启桶默认加密。
- **功能限制：**
  - 只支持生成CSV格式的清单文件。
  - 桶清单筛选条件目前仅支持设置为所有对象或指定前缀的对象。
  - 同一个桶中多条清单规则的筛选条件不能彼此包含：
    - 如果已经存在针对桶中所有对象的规则，则无法再创建按对象名前缀筛选的规则。如需创建，要先删除针对所有对象的规则。
    - 如果已经存在按对象名前缀筛选的规则，则无法再创建针对桶中所有对象的规则。如需创建，要先删除所有按对象名前缀筛选的规则。

- 如果已经存在某个按对象名前缀筛选的规则（如前缀ab），则无法再创建与其存在包含或被包含关系的规则（如前缀a或前缀abc）。如需创建，要先删除存在包含或被包含关系的规则。
- 桶清单加密方式目前只支持SSE-KMS。
- **权限限制：**
  - 清单文件使用OBS系统用户上传到目标桶，目标桶必须给OBS系统用户写权限。
- **其他：**
  - 暂不对桶清单功能收费，桶清单生成后只按照存量计费。

## 方法定义

obsClient.setInventoryConfiguration([SetInventoryConfigurationRequest request](#))

## 请求参数说明

表 7-80 SetInventoryConfigurationRequest

参数名称	参数类型	是否必选	描述
bucketName	String	是	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"><li>• 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。</li><li>• 桶命名规则如下：<ul style="list-style-type: none"><li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>- 禁止使用IP地址。</li><li>- 禁止以“-”或“.”开头及结尾。</li><li>- 禁止两个“.”相邻（如：“my.bucket”）。</li><li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>• 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。</li></ul> <p><b>默认取值：</b> 无</p>

参数名称	参数类型	是否必选	描述
inventoryConfiguration	<b>Inventory Configuration</b>	是	<b>参数解释:</b> 桶清单配置规则, 详见 <a href="#">InventoryConfiguration</a> 。

表 7-81 InventoryConfiguration

参数名称	参数类型	是否必选	描述
configurationId	String	必选	<b>参数解释:</b> 桶清单配置规则的id。 <b>约束限制:</b> 必须由以下字符组成"a-z"、"A-Z"、"0-9"、"-","_"和"."。 <b>取值范围:</b> 长度最长为64的字符串。 <b>默认取值:</b> 无
isEnabled	boolean	必选	<b>参数解释:</b> 桶清单配置规则是否启用。 <b>取值范围:</b> true: 生成清单文件。 false: 不生成清单文件。 <b>默认取值:</b> true
objectPrefix	String	可选	<b>参数解释:</b> 前缀过滤条件, 清单文件中只生成以此前缀开头的对象列表。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无



参数名称	参数类型	是否必选	描述
frequency	String	必选	<p><b>参数解释:</b> 清单文件的生成周期，只支持按天和按周生成清单，第一次配置完桶清单，任务会在一个小时内启动，之后每隔一个周期启动一次。</p> <p><b>取值范围:</b> Daily: 按天生成清单 Weekly: 按周生成清单</p> <p><b>默认取值:</b> 无</p>
inventoryFormat	String	必选	<p><b>参数解释:</b> 生成的清单文件的格式，现只支持CSV格式。</p> <p><b>取值范围:</b> CSV</p> <p><b>默认取值:</b> 无</p>
destinationBucket	String	必选	<p><b>参数解释:</b> 存放清单文件的目标桶的桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
inventoryPrefix	String	可选	<p><b>参数解释:</b> 生成的清单文件对象名会以此前缀开头。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> BucketInventory, 即如果不配置前缀, 则生成的清单文件对象名默认以"BucketInventory"字符串为前缀开头。</p>
includedObjectVersions	String	必选	<p><b>参数解释:</b> 清单文件中包含对象的多版本配置。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>All: 清单会包含对象所有的版本, 清单中会增加版本相关的字段: VersionId、IsLatest和DeleteMarker。</li> <li>Current: 清单文件中只会列出当前版本信息, 不会出现版本相关字段。</li> </ul> <p><b>默认取值:</b> 无</p>
optionalFields	ArrayList<String>	可选	<p><b>参数解释:</b> 在此选项中可以添加一些额外的对象元数据字段, 生成的清单文件中会包含optionalFields中配置的字段。</p> <p><b>取值范围:</b></p> <p>Size: 对象大小。 LastModifiedDate: 对象最后修改时间。 StorageClass: 对象的存储类别。 ETag: 对象的ETag值。 IsMultipartUploaded: 对象是否为<a href="#">多段上传</a>。 ReplicationStatus: 对象的跨区域复制状态。 EncryptionStatus: 对象的加密状态。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 7-82 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例将为example-bucket桶设置桶清单，桶清单会统计前缀为exampleObjectPrefix的对象的元数据。清单文件将存储在example-target-bucket桶中，清单文件的前缀为exampleInventoryPrefix。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HeaderResponse;
import com.obs.services.model.inventory.InventoryConfiguration;
import com.obs.services.model.inventory.SetInventoryConfigurationRequest;
public class SetInventoryConfiguration001
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        // String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);
    }
}
```

```
try {
    // 设置相关示例参数
    String exampleBucketName = "example-bucket";
    String exampleTargetBucketName = "example-target-bucket";
    String exampleConfigurationId = "exampleConfigId001";
    String exampleInventoryPrefix = "exampleInventoryPrefix";
    String exampleObjectPrefix = "exampleObjectPrefix";
    // 设置桶清单配置规则详细参数
    InventoryConfiguration exampleConfiguration = new InventoryConfiguration();
    exampleConfiguration.setDestinationBucket(exampleTargetBucketName);
    exampleConfiguration.setConfigurationId(exampleConfigurationId);
    exampleConfiguration.setInventoryFormat(InventoryConfiguration.InventoryFormatOptions.CSV);
    exampleConfiguration.setFrequency(InventoryConfiguration.FrequencyOptions.DAILY);
    exampleConfiguration.setEnabled(true);
    exampleConfiguration.setIncludedObjectVersions(
        InventoryConfiguration.IncludedObjectVersionsOptions.CURRENT);
    exampleConfiguration.setInventoryPrefix(exampleInventoryPrefix);
    exampleConfiguration.setObjectPrefix(exampleObjectPrefix);
    // 设置清单文件中会包含的额外的对象元数据字段
    exampleConfiguration
        .getOptionalFields()
        .add(InventoryConfiguration.OptionalFieldOptions.IS_MULTIPART_UPLOADED);
    exampleConfiguration.getOptionalFields().add(InventoryConfiguration.OptionalFieldOptions.ETAG);
    exampleConfiguration
        .getOptionalFields()
        .add(InventoryConfiguration.OptionalFieldOptions.REPLICATION_STATUS);
    SetInventoryConfigurationRequest request =
        new SetInventoryConfigurationRequest(exampleBucketName, exampleConfiguration);
    // 设置桶清单配置规则
    HeaderResponse response = obsClient.setInventoryConfiguration(request);
    System.out.println("SetInventoryConfiguration succeeded");
    System.out.println("HTTP Code: " + response.getStatusCode());
} catch (ObsException e) {
    System.out.println("SetInventoryConfiguration failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code: " + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
} catch (Exception e) {
    System.out.println("SetInventoryConfiguration failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置桶清单的API说明, 请参见[设置桶清单](#)。
- 设置桶清单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](#)。

## 7.17.2 获取桶清单规则(Java SDK)

### 功能说明

您可以使用获取桶清单规则接口获取指定桶的清单规则, 可以通过配置规则id来选择指定清单规则。

## 接口约束

- 您必须是桶所有者或者拥有获取桶清单权限，才能获取桶清单。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs.bucket:GetInventoryConfiguration权限，如果使用桶策略则需授予GetInventoryConfiguration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以及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.getInventoryConfiguration([GetInventoryConfigurationRequest](#)  
[request](#))

## 请求参数说明

表 7-83 GetInventoryConfigurationRequest

参数名称	参数类型	是否必选	描述
bucketName	String	是	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"><li>• 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。</li><li>• 桶命名规则如下：<ul style="list-style-type: none"><li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>- 禁止使用IP地址。</li><li>- 禁止以“-”或“.”开头及结尾。</li><li>- 禁止两个“.”相邻（如：“my.bucket”）。</li><li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>• 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。</li></ul> <p><b>默认取值：</b> 无</p>

参数名称	参数类型	是否必选	描述
configurationId	String	是	<b>参数解释:</b> 清单配置的id。 <b>约束限制:</b> 最长64字节,必须由以下字符组成"a-z"、"A-Z"、"0-9"、"-","_"和"." <b>默认取值:</b> 无

## 返回结果说明

表 7-84 GetInventoryConfigurationResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
inventoryConfiguration	<a href="#">InventoryConfiguration</a>	<b>参数解释:</b> 桶清单配置规则，详见 <a href="#">InventoryConfiguration</a> 。

表 7-85 InventoryConfiguration

参数名称	参数类型	是否必选	描述
configurationId	String	必选	<p><b>参数解释:</b> 桶清单配置规则的id。</p> <p><b>约束限制:</b> 必须由以下字符组成"a-z"、"A-Z"、"0-9"、"-","_"和"."。</p> <p><b>取值范围:</b> 长度最长为64的字符串。</p> <p><b>默认取值:</b> 无</p>
isEnabled	boolean	必选	<p><b>参数解释:</b> 桶清单配置规则是否启用。</p> <p><b>取值范围:</b> true: 生成清单文件。 false: 不生成清单文件。</p> <p><b>默认取值:</b> true</p>
objectPrefix	String	可选	<p><b>参数解释:</b> 前缀过滤条件, 清单文件中只生成以此前缀开头的对象列表。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
frequency	String	必选	<p><b>参数解释:</b> 清单文件的生成周期, 只支持按天和按周生成清单, 第一次配置完桶清单, 任务会在一个小时内启动, 之后每隔一个周期启动一次。</p> <p><b>取值范围:</b> Daily: 按天生成清单 Weekly: 按周生成清单</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
inventoryFormat	String	必选	<b>参数解释:</b> 生成的清单文件的格式，现只支持CSV格式。 <b>取值范围:</b> CSV <b>默认取值:</b> 无
destinationBucket	String	必选	<b>参数解释:</b> 存放清单文件的目标桶的桶名。 <b>约束限制:</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <b>默认取值:</b> 无
inventoryPrefix	String	可选	<b>参数解释:</b> 生成的清单文件对象名会以此前缀开头。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> BucketInventory，即如果不配置前缀，则生成的清单文件对象名默认以“BucketInventory”字符串为前缀开头。



参数名称	参数类型	是否必选	描述
includedObjectVersions	String	必选	<p><b>参数解释:</b> 清单文件中包含对象的多版本配置。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>All: 清单会包含对象所有的版本, 清单中会增加版本相关的字段: VersionId、IsLatest和DeleteMarker。</li> <li>Current: 清单文件中只会列出当前版本信息, 不会出现版本相关字段。</li> </ul> <p><b>默认取值:</b> 无</p>
optionalFields	ArrayList<String>	可选	<p><b>参数解释:</b> 在此选项中可以添加一些额外的对象元数据字段, 生成的清单文件中会包含optionalFields中配置的字段。</p> <p><b>取值范围:</b></p> <p>Size: 对象大小。</p> <p>LastModifiedDate: 对象最后修改时间。</p> <p>StorageClass: 对象的存储类别。</p> <p>ETag: 对象的ETag值。</p> <p>IsMultipartUploaded: 对象是否为<a href="#">多段上传</a>。</p> <p>ReplicationStatus: 对象的跨区域复制状态。</p> <p>EncryptionStatus: 对象的加密状态。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于获取example-bucket桶, 配置规则id为exampleConfigId001的清单规则。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.inventory.GetInventoryConfigurationRequest;
import com.obs.services.model.inventory.GetInventoryConfigurationResult;
import com.obs.services.model.inventory.InventoryConfiguration;
public class GetInventoryConfiguration001
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK, 也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken, 也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint, 此处以华北-北京四为例, 其他地区请按实际情况填写。
    }
}
```

```
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
// String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 设置相关示例参数
    String exampleBucketName = "example-bucket";
    String exampleConfigurationId = "exampleConfigId001";
    GetInventoryConfigurationRequest request =
        new GetInventoryConfigurationRequest(exampleBucketName, exampleConfigurationId);
    // 获取桶清单配置规则
    GetInventoryConfigurationResult result = obsClient.getInventoryConfiguration(request);
    InventoryConfiguration inventoryConfiguration = result.getInventoryConfiguration();
    // 打印inventoryConfiguration的所有成员
    System.out.println("ConfigurationId:" + inventoryConfiguration.getConfigurationId());
    System.out.println("DestinationBucket:" + inventoryConfiguration.getDestinationBucket());
    System.out.println("InventoryFormat:" + inventoryConfiguration.getInventoryFormat());
    System.out.println("Enabled:" + inventoryConfiguration.getEnabled());
    System.out.println("Frequency:" + inventoryConfiguration.getFrequency());
    System.out.println("IncludedObjectVersions:" + inventoryConfiguration.getIncludedObjectVersions());
    System.out.println("InventoryPrefix:" + inventoryConfiguration.getInventoryPrefix());
    System.out.println("ObjectPrefix:" + inventoryConfiguration.getObjectPrefix());
    System.out.println("OptionalFields:" + inventoryConfiguration.getOptionalFields());
    // 打印HTTP状态码
    System.out.println("HTTP Code: " + result.getStatusCode());
    System.out.println("GetInventoryConfiguration succeeded");
} catch (ObsException e) {
    System.out.println("GetInventoryConfiguration failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code: " + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
} catch (Exception e) {
    System.out.println("GetInventoryConfiguration failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置桶清单的API说明，请参见[获取桶清单](#)。
- 设置桶清单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](#)。

## 7.17.3 列举桶清单规则(Java SDK)

### 功能说明

OBS使用不带清单id的GET操作来获取指定桶的所有清单配置，获取到的清单配置一次性返回，不分页。

## 接口约束

- 您必须是桶所有者或者拥有列举桶清单权限，才能列举桶清单。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:GetInventoryConfiguration权限，如果使用桶策略则需授予GetInventoryConfiguration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以及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.listInventoryConfiguration([ListInventoryConfigurationRequest request](#))

## 请求参数说明

表 7-86 ListInventoryConfigurationRequest

参数名称	参数类型	是否必选	描述
bucketName	String	是	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"><li>• 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。</li><li>• 桶命名规则如下：<ul style="list-style-type: none"><li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。</li><li>- 禁止使用IP地址。</li><li>- 禁止以“-”或“.”开头及结尾。</li><li>- 禁止两个“.”相邻（如：“my..bucket”）。</li><li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>• 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。</li></ul> <p><b>默认取值：</b> 无</p>

## 返回结果说明

表 7-87 ListInventoryConfigurationResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
inventoryConfigurations	List< <a href="#">InventoryConfiguration</a> >	<b>参数解释:</b> 桶的所有桶清单配置规则，详见 <a href="#">InventoryConfiguration</a>

表 7-88 InventoryConfiguration

参数名称	参数类型	是否必选	描述
configurationId	String	必选	<b>参数解释:</b> 桶清单配置规则的id。 <b>约束限制:</b> 必须由以下字符组成"a-z"、"A-Z"、"0-9"、"-","_"和"."。 <b>取值范围:</b> 长度最长为64的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
isEnabled	boolean	必选	<p><b>参数解释:</b> 桶清单配置规则是否启用。</p> <p><b>取值范围:</b> true: 生成清单文件。 false: 不生成清单文件。</p> <p><b>默认取值:</b> true</p>
objectPrefix	String	可选	<p><b>参数解释:</b> 前缀过滤条件, 清单文件中只生成以此前缀开头的对象列表。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
frequency	String	必选	<p><b>参数解释:</b> 清单文件的生成周期, 只支持按天和按周生成清单, 第一次配置完桶清单, 任务会在一个小时内启动, 之后每隔一个周期启动一次。</p> <p><b>取值范围:</b> Daily: 按天生成清单 Weekly: 按周生成清单</p> <p><b>默认取值:</b> 无</p>
inventoryFormat	String	必选	<p><b>参数解释:</b> 生成的清单文件的格式, 现只支持CSV格式。</p> <p><b>取值范围:</b> CSV</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
destinationBucket	String	必选	<p><b>参数解释:</b> 存放清单文件的目标桶的桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
inventoryPrefix	String	可选	<p><b>参数解释:</b> 生成的清单文件对象名会以此前缀开头。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> BucketInventory，即如果不配置前缀，则生成的清单文件对象名默认以"BucketInventory"字符串为前缀开头。</p>
includedObjectVersions	String	必选	<p><b>参数解释:</b> 清单文件中包含对象的多版本配置。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>All：清单会包含对象所有的版本，清单中会增加版本相关的字段：VersionId、IsLatest和DeleteMarker。</li> <li>Current：清单文件中只会列出当前版本信息，不会出现版本相关字段。</li> </ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
optionalFields	ArrayList<String>	可选	<p><b>参数解释:</b> 在此选项中可以添加一些额外的对象元数据字段，生成的清单文件中会包含optionalFields中配置的字段。</p> <p><b>取值范围:</b> Size: 对象大小。 LastModifiedDate: 对象最后修改时间。 StorageClass: 对象的存储类别。 ETag: 对象的ETag值。 IsMultipartUploaded: 对象是否为<a href="#">多段上传</a>。 ReplicationStatus: 对象的跨区域复制状态。 EncryptionStatus: 对象的加密状态。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于列举example-bucket桶的所有桶清单文件。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.inventory.InventoryConfiguration;
import com.obs.services.model.inventory.ListInventoryConfigurationRequest;
import com.obs.services.model.inventory.ListInventoryConfigurationResult;
import java.util.List;
public class ListInventoryConfigurations001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        // String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 设置相关示例参数
            String exampleBucketName = "example-bucket";
            ListInventoryConfigurationRequest request = new
            ListInventoryConfigurationRequest(exampleBucketName);
            // 列举桶清单配置规则
```

```
ListInventoryConfigurationResult result = obsClient.listInventoryConfiguration(request);
List<InventoryConfiguration> inventoryConfigurations = result.getInventoryConfigurations();
// 打印inventoryConfigurations的所有规则的所有成员
for (InventoryConfiguration inventoryConfiguration : inventoryConfigurations) {
    System.out.println("ConfigurationId:" + inventoryConfiguration.getConfigurationId());
    System.out.println("DestinationBucket:" + inventoryConfiguration.getDestinationBucket());
    System.out.println("InventoryFormat:" + inventoryConfiguration.getInventoryFormat());
    System.out.println("Enabled:" + inventoryConfiguration.getEnabled());
    System.out.println("Frequency:" + inventoryConfiguration.getFrequency());
    System.out.println("IncludedObjectVersions:" +
inventoryConfiguration.getIncludedObjectVersions());
    System.out.println("InventoryPrefix:" + inventoryConfiguration.getInventoryPrefix());
    System.out.println("ObjectPrefix:" + inventoryConfiguration.getObjectPrefix());
    System.out.println("OptionalFields:" + inventoryConfiguration.getOptionalFields());
    System.out.println("-----");
}
// 打印HTTP状态码
System.out.println("HTTP Code: " + result.getStatusCode());
System.out.println("ListInventoryConfiguration succeeded");
} catch (ObsException e) {
    System.out.println("ListInventoryConfiguration failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code: " + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
} catch (Exception e) {
    System.out.println("ListInventoryConfiguration failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置桶清单的API说明，请参见[列举桶清单](#)。
- 设置桶清单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](#)。

## 7.17.4 删除桶清单规则(Java SDK)

### 功能说明

OBS使用DELETE操作来删除指定桶的清单配置（通过清单规则id来指定确认要删除的桶清单规则）。

### 接口约束

- 您必须是桶拥有者或者拥有获取桶清单权限，才能获取桶清单。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket:DeleteBucketInventoryConfiguration权限，如果使用桶策略则需授予DeleteBucketInventoryConfiguration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以及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.deleteInventoryConfiguration([DeleteInventoryConfigurationRequest request](#))

## 请求参数说明

表 7-89 DeleteInventoryConfigurationRequest

参数名称	参数类型	是否必选	描述
bucketName	String	是	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值：</b> 无</p>
configurationId	String	是	<p><b>参数解释：</b> 要删除的清单配置的id。最长64字节,必须由以下字符组成"a-z"、"A-Z"、"0-9"、“-”、“_”和“.”</p> <p><b>默认取值：</b> 无</p>

## 返回结果说明

表 7-90 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于删除example-bucket桶的exampleConfigId001清单规则。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HeaderResponse;
import com.obs.services.model.inventory.DeletelInventoryConfigurationRequest;
import com.obs.services.model.inventory.GetInventoryConfigurationRequest;
import com.obs.services.model.inventory.GetInventoryConfigurationResult;
import com.obs.services.model.inventory.InventoryConfiguration;
public class DeletelInventoryConfiguration001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        // String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);
    }
}
```

```
try {
    // 设置相关示例参数
    String exampleBucketName = "example-bucket";
    String exampleConfigurationId = "exampleConfigId001";
    DeleteInventoryConfigurationRequest request =
        new DeleteInventoryConfigurationRequest(exampleBucketName, exampleConfigurationId);
    // 删除桶清单配置规则
    HeaderResponse result = obsClient.deleteInventoryConfiguration(request);
    // 打印HTTP状态码
    System.out.println("HTTP Code: " + result.getStatusCode());
    System.out.println("DeleteInventoryConfiguration succeeded");
} catch (ObsException e) {
    System.out.println("DeleteInventoryConfiguration failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code: " + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
} catch (Exception e) {
    System.out.println("DeleteInventoryConfiguration failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置桶清单的API说明, 请参见[删除桶清单](#)。
- 设置桶清单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](#)。

# 8 管理并行文件系统(Java SDK)

您可以参考[并行文件系统简介](#)来了解什么是并行文件系统。关于并行文件系统支持的API可参考[并行文件系统支持的API列表](#)，本章节主要介绍并行文件系统不同于对象桶的操作。

## 8.1 创建并行文件系统(Java SDK)

### 功能说明

并行文件系统（Parallel File System）是对象存储服务（Object Storage Service，OBS）提供的一种经过优化的高性能文件系统，提供毫秒级别访问时延，TB/s级别带宽和百万级别的IOPS，能够快速处理高性能计算（HPC）工作负载。

作为对象存储服务的子产品，并行文件系统支持用户按照标准的OBS接口读取数据。也支持通过部署在弹性云服务器中的PFS客户端（obsfs工具），按照POSIX文件语义读写数据；通过obsfs用户可以将创建的并行文件系统挂载到云端Linux服务器上，并能像操作本地文件系统一样对并行文件系统内的文件和目录进行在线处理，包括：创建和删除文件/目录，重命名文件/目录，修改写文件等操作。

您可以通过ObsClient.createBucket创建并行文件系统。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 您必须拥有obs:bucket:CreateBucket权限，才能创建并行文件系统。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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 不支持将已有的OBS桶修改为并行文件系统。
- 并行文件系统暂不支持配额配置功能，默认无配额限制。

- 同一账号下，可以创建多个桶，数量上限是100个（包括对象桶和并行文件系统，不区分地域），存储桶中的对象数量和大小没有限制。
- 新建并行文件系统的桶名在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。
- 用户删除并行文件系统后，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并行文件系统，你可以在[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([CreateBucketRequest request](#))

## 请求参数说明

表 8-1 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">CreateBucketRequest</a>	是	<b>参数解释：</b> 创建并行文件系统相关信息。详见 <a href="#">CreateBucketRequest</a> 。

表 8-2 CreateBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 并行文件系统的桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>同一用户在同一个区域多次创建同名并行文件系统不会报错，创建的并行文件系统属性以第一次请求为准。</li> </ul> <p><b>默认取值:</b> 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><b>参数解释:</b> 并行文件系统所在的区域。</p> <p><b>约束限制:</b> 该参数定义了并行文件系统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><b>取值范围:</b> 当前有效的OBS区域和终端节点的更多信息，请参考<a href="#">地区和终端节点</a>。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><b>默认取值:</b> 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	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b>                      创并行文件系统时可指定并行文件系统的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b>                      AccessControlList.REST_CANNED_PRIVATE</p>
storageClasses	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b>                      创建并行文件系统时暂不支持配置并行文件系统默认存储类别。您可以通过<a href="#">设置桶存储类别(Java SDK)</a>接口配置并行文件系统存储类别。</p> <p><b>默认取值:</b>                      无</p>
extensionPermissionMap	Map< <a href="#">ExtensionBucketPermissionEnum</a> , Set<String>>	可选	<p><b>参数解释:</b>                      并行文件系统ACL的授权Map，您可以为一个或多个账号授予并行文件系统权限。Map的<a href="#">ExtensionBucketPermissionEnum</a>用于指定权限，Map的Set&lt;String&gt;用于说明该权限授予的账号ID列表，即domain_id列表。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 授予权限的范围详见<a href="#">ExtensionBucketPermissionEnum</a>。</li> <li>• 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a></li> </ul> <p><b>默认取值:</b>                      无</p>

参数名称	参数类型	是否必选	描述
epid	String	可选	<p><b>参数解释:</b> 创并行文件系统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><b>约束限制:</b> 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><b>默认取值:</b> 无</p>
availableZone	AvailableZoneEnum	可选	<p><b>参数解释:</b> 创并行文件系统时可指定的数据冗余类型, 即AZ类型。</p> <p><b>约束限制:</b> 归档存储不支持多AZ。如果并行文件系统所在区域不支持多AZ存储, 则该并行文件系统的存储类别默认为单AZ。</p> <p><b>取值范围:</b> 如果将并行文件系统配置为多AZ, 则取值为"MULTI_AZ"。如果将并行文件系统配置为单AZ, 则无需选择此选项, 系统默认赋值。</p> <p><b>默认取值:</b> 不设置该参数, 则默认设置并行文件系统为单AZ。</p>
bucketType	BucketTypeEnum	可选	<p><b>参数解释:</b> 创建的并行文件系统类型。</p> <p><b>取值范围:</b> 创建并行文件系统时, bucketType必须选择PFS。</p> <p><b>默认取值:</b> 无</p>

表 8-3 ACL 预定义访问策略

常量名	说明
AccessControlList.REST_CANNED_PRIVATE	<p>私有读写。</p> <p>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。</p>



常量名	说明
AccessControlList.REST_CANNED_PUBLIC_READ	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。 设在对象上，所有人可以获取该对象内容和元数据。
AccessControlList.REST_CANNED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControlList.REST_CANNED_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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 8-4 ExtensionBucketPermissionEnum

常量名	描述
GRANT_READ	授予对应账号ID的账号READ权限。 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。
GRANT_WRITE	授予对应账号ID的账号WRITE权限。 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、拷贝段、合并段、取消多段上传任务。
GRANT_READ_ACP	授予对应账号ID的账号READ_ACP权限。 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。

常量名	描述
GRANT_WRITE_ACP	授予对应账号ID的账号WRITE_ACP权限。 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。
GRANT_FULL_CONTROL	授予对应账号ID的账号FULL_CONTROL权限。 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。
GRANT_READ_DELIVERED	授予对应账号ID的账号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。
GRANT_FULL_CONTROL_DELIVERED	授予对应账号ID的账号FULL_CONTROL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。

表 8-5 AccessControlList

参数名称	参数类型	是否必选	描述
owner	<b>Owner</b>	可选	<b>参数解释：</b> 并行文件系统所有者的信息，详见 <b>Owner</b> 。
delivered	boolean	可选	<b>参数解释：</b> 并行文件系统的ACL是否向并行文件系统内对象传递，作用于并行文件系统内所有对象。 <b>取值范围：</b> true：是，并行文件系统ACL向并行文件系统内对象传递。 false：否，并行文件系统ACL不向并行文件系统内对象传递，仅作用于并行文件系统。 <b>默认取值：</b> false
grants	Set< <b>GrantAndPermission</b> >	可选	<b>参数解释：</b> 被授权用户相关信息，详见 <b>GrantAndPermission</b> 。

表 8-6 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 8-7 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<p><b>参数解释:</b> 被授权用户或用户组, 详见<a href="#">GranteeInterface</a>。</p>
permission	<a href="#">Permission</a>	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
delivered	boolean	可选	<p><b>参数解释:</b> 并行文件系统的ACL是否向并行文件系统内对象传递，作用于并行文件系统内所有对象。</p> <p><b>取值范围:</b> true: 是，并行文件系统ACL向并行文件系统内对象传递。 false: 否，并行文件系统ACL不向并行文件系统内对象传递，仅作用于并行文件系统。</p> <p><b>默认取值:</b> false</p>

表 8-8 GranteeInterface

参数名称	参数类型	是否必选	描述
<b>CanonicalGrantee</b>	<b>CanonicalGrantee</b>	必选	<p><b>参数解释:</b> 被授权用户的用户信息，详见<b>CanonicalGrantee</b>。</p>
<b>GroupGrantee</b>	<b>GroupGrantee</b>	必选	<p><b>参数解释:</b> 被授权用户组的用户组信息。</p> <p><b>取值范围:</b> 详见<b>GroupGrantee</b>。</p> <p><b>默认取值:</b> 无</p>

表 8-9 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<p><b>参数解释:</b> 被授权用户的账号ID，即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a></p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
displayName	String	可选	<p><b>参数描述:</b> 被授权用户的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 8-10 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户，已废弃。
LOG_DELIVERY	日志投递组，已废弃。

表 8-11 Permission

常量名	原始值	说明
PERMISSION_READ	READ	<p>读权限。</p> <p>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。</p> <p>如果有对象的读权限，则可以获取该对象内容和元数据。</p>
PERMISSION_WRITE	WRITE	<p>写权限。</p> <p>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。</p> <p>此权限在对象上不适用。</p>
PERMISSION_READ_ACP	READ_ACP	<p>读取ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有读对应桶或对象ACP的权限。</p>

常量名	原始值	说明
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

表 8-12 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 8-13 AvailableZoneEnum

常量名	原始值	说明
MULTI_AZ	3az	多az类型

## 返回结果说明

表 8-14 response

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
bucketName	String	<p><b>参数解释:</b> 并行文件系统名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>并行文件系统的名字需全局唯一，不能与已有的任何并行文件系统名称重复，包括其他用户创建的并行文件系统。</li><li>并行文件系统命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>同一用户在同一个区域多次创建同名并行文件系统不会报错，创建的并行文件系统属性以第一次请求为准。</li></ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
owner	<b>Owner</b>	<b>参数解释:</b> 并行文件系统所有者信息, 详见 <b>Owner</b> 。
creationDate	java.util.Date	<b>参数解释:</b> 并行文件系统创建时间。 <b>默认取值:</b> 无
location	String	<b>参数解释:</b> 并行文件系统所在的区域。 <b>约束限制:</b> 该参数定义了并行文件系统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。 <b>取值范围:</b> 当前有效的OBS区域和终端节点的更多信息, 请参考 <b>地区和终端节点</b> 。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。 <b>默认取值:</b> 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)。
storageClass	<b>StorageClassEnum</b>	<b>参数解释:</b> 创建并行文件系统时暂不支持配置并行文件系统默认存储类别。您可以通过 <b>设置桶存储类别(Java SDK)</b> 接口配置并行文件系统存储类别。 <b>默认取值:</b> 无



参数名称	参数类型	描述
acl	<a href="#">AccessControlList</a>	<p><b>参数解释:</b> 创并行文件系统时可指定并行文件系统的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List，ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>
bucketTypeEnum	BucketTypeEnum	<p><b>参数解释:</b> 创建的并行文件系统类型。</p> <p><b>取值范围:</b> 创建并行文件系统时，bucketType为PFS。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于新建名为examplebucket的并行文件系统。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketTypeEnum;
import com.obs.services.model.CreateBucketRequest;
import com.obs.services.model.HeaderResponse;
public class CreateBucket001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);
```

```
try {
    // 创建并行文件系统成功
    CreateBucketRequest request = new CreateBucketRequest();
    request.setBucketName("examplebucket");
    request.setBucketType(BucketTypeEnum.PFS);
    // 指定所属区域, location 需要与 endpoint的位置信息 一致
    request.setLocation("your_region");
    HeaderResponse response = obsClient.createBucket(request);
    System.out.println("CreateBucket successfully");
    System.out.println("StatusCode:" + response.getStatusCode());
    System.out.println("RequestId:" + response.getRequestId());
} catch (ObsException e) {
    System.out.println("CreateBucket failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("CreateBucket failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于并行文件系统的更多说明, 请参见[什么是并行文件系统](#)。
- 更多关于创建并行文件系统的示例代码, 请参见[Github示例](#)。
- 创建并行文件系统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](#)。
- 创建并行文件系统常见问题请参见[并行文件系统常见问题](#)。

## 8.2 列举并行文件系统(Java SDK)

### 功能说明

并行文件系统 (Parallel File System) 是对象存储服务 (Object Storage Service, OBS) 提供的一种经过优化的高性能文件系统, 提供毫秒级别访问时延, TB/s级别带宽和百万级别的IOPS, 能够快速处理高性能计算 (HPC) 工作负载。

作为对象存储服务的子产品, 并行文件系统支持用户按照标准的OBS接口读取数据。也支持通过部署在弹性云服务器中的PFS客户端 (obsfs工具), 按照POSIX文件语义读写数据; 通过obsfs用户可以将创建的并行文件系统挂载到云端Linux服务器上, 并能像操作本地文件系统一样对并行文件系统内的文件和目录进行在线处理, 包括: 创建和删除文件/目录, 重命名文件/目录, 修改写文件等操作。

您可以通过ObsClient.listBuckets列举已存在的并行文件系统。

**须知**

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

**接口约束**

- 您必须拥有obs:bucket:ListAllMyBuckets权限，才能获取并行文件系统列表。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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 并行文件系统暂不支持配额配置功能，默认无配额限制。
- 必须设置ListBucketsRequest.setBucketType参数为BucketTypeEnum.PFS，才可以列举出当前用户的并行文件系统。
- 设置ListBucketsRequest.setQueryLocation参数为true后，可在列举并行文件系统时查询并行文件系统的区域位置。

**方法定义**

obsClient.listBuckets([ListBucketsRequest request](#))

**请求参数说明****表 8-15** 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">ListBucketsRequest</a>	必选	<b>参数解释：</b> 获取并行文件系统列表请求参数，详见 <a href="#">ListBucketsRequest</a> 。

**表 8-16** ListBucketsRequest

参数名称	参数类型	是否必选	描述
queryLocation	boolean	可选	<b>参数解释：</b> 是否同时查询并行文件系统的区域位置。 <b>取值范围：</b> <ul style="list-style-type: none"><li>• true: 同时查询并行文件系统的区域位置。</li><li>• false: 不查询并行文件系统的区域位置。</li></ul> <b>默认取值：</b> false

参数名称	参数类型	是否必选	描述
bucketType	BucketTypeEnum	可选	<p><b>参数解释:</b> 列举指定类型的并行文件系统。</p> <p><b>取值范围:</b> 列举并行文件系统时，bucketType必须选择PFS。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 8-17 ListBucketsResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
owner	<a href="#">Owner</a>	<p><b>参数解释:</b> 并行文件系统所有者信息，详见<a href="#">Owner</a>。</p> <p><b>默认取值:</b> 无</p>
buckets	List< <a href="#">ObsBucket</a> >	<p><b>参数解释:</b> 并行文件系统列表信息。详见<a href="#">ObsBucket</a>。</p> <p><b>默认取值:</b> 无</p>

表 8-18 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 8-19 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 8-20 ACL 预定义访问策略

常量名	说明
AccessControlList.REST_CANNED_PRIVATE	<p>私有读写。</p> <p>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。</p>
AccessControlList.REST_CANNED_PUBLIC_READ	<p>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。</p> <p>设在对象上, 所有人可以获取该对象内容和元数据。</p>

常量名	说明
AccessControllist.REST_CAN NED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControllist.REST_CAN NED_PUBLIC_READ_DELIVER ED	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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControllist.REST_CAN NED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 8-21 AccessControllist

参数名称	参数类型	是否必选	描述
owner	<b>Owner</b>	可选	<b>参数解释：</b> 并行文件系统所有者的信息，详见 <b>Owner</b> 。
delivered	boolean	可选	<b>参数解释：</b> 并行文件系统的ACL是否向并行文件 系统内对象传递，作用于并行文件 系统内所有对象。 <b>取值范围：</b> true：是，并行文件系统ACL向并行 文件系统内对象传递。 false：否，并行文件系统ACL不向并 行文件系统内对象传递，仅作用于并 行文件系统。 <b>默认取值：</b> false
grants	Set< <b>GrantAndPer mission</b> >	可选	<b>参数解释：</b> 被授权用户相关信息，详见 <b>GrantAndPermission</b> 。

表 8-22 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组, 详见 <a href="#">GranteeInterface</a> 。
permission	<a href="#">Permission</a>	必选	<b>参数解释:</b> 用户或用户组被授予的权限。 <b>取值范围:</b> 详见 <a href="#">Permission</a> 。 <b>默认取值:</b> 无
delivered	boolean	可选	<b>参数解释:</b> 并行文件系统的ACL是否向并行文件系统内对象传递, 作用于并行文件系统内所有对象。 <b>取值范围:</b> true: 是, 并行文件系统ACL向并行文件系统内对象传递。 false: 否, 并行文件系统ACL不向并行文件系统内对象传递, 仅作用于并行文件系统。 <b>默认取值:</b> false

表 8-23 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 8-24 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<p><b>参数解释:</b> 被授权用户的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
displayName	String	可选	<p><b>参数描述:</b> 被授权用户的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 8-25 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 8-26 Permission

常量名	原始值	说明
PERMISSION_READ	READ	<p>读权限。</p> <p>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。</p> <p>如果有对象的读权限, 则可以获取该对象内容和元数据。</p>
PERMISSION_WRITE	WRITE	<p>写权限。</p> <p>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。</p> <p>此权限在对象上不适用。</p>



常量名	原始值	说明
PERMISSION_READ_ACP	READ_ACP	<p>读取ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有读对应桶或对象ACP的权限。</p>
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

表 8-27 ObsBucket

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释：</b> HTTP状态码。</p> <p><b>取值范围：</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值：</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
bucketName	String	<p><b>参数解释:</b> 并行文件系统名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>并行文件系统的名字需全局唯一，不能与已有的任何并行文件系统名称重复，包括其他用户创建的并行文件系统。</li> <li>并行文件系统命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>同一用户在同一个区域多次创建同名并行文件系统不会报错，创建的并行文件系统属性以第一次请求为准。</li> </ul> <p><b>默认取值:</b> 无</p>
owner	<b>Owner</b>	<p><b>参数解释:</b> 并行文件系统拥有者信息，详见<a href="#">Owner</a>。</p>
creationDate	java.util.Date	<p><b>参数解释:</b> 并行文件系统创建时间。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
location	String	<p><b>参数解释:</b> 并行文件系统所在的区域。</p> <p><b>约束限制:</b> 该参数定义了并行文件系统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><b>取值范围:</b> 当前有效的OBS区域和终端节点的更多信息，请参考<a href="#">地区和终端节点</a>。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><b>默认取值:</b> 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>
storageClass	StorageClassEnum	<p><b>参数解释:</b> 创建并行文件系统时暂不支持配置并行文件系统默认存储类别。您可以通过<a href="#">设置桶存储类别(Java SDK)</a>接口配置并行文件系统存储类别。</p> <p><b>默认取值:</b> 无</p>
acl	AccessControlList	<p><b>参数解释:</b> 创并行文件系统时可指定并行文件系统的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List，ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>

参数名称	参数类型	描述
bucketTypeEnum	BucketTypeEnum	<p><b>参数解释:</b> 创建的并行文件系统类型。</p> <p><b>取值范围:</b> 列举并行文件系统时，bucketType为PFS。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于列举并行文件系统的信息。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketTypeEnum;
import com.obs.services.model.ListBucketsRequest;
import com.obs.services.model.ObsBucket;
import java.util.List;
public class ListBucket001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 列举并行文件系统列表
            ListBucketsRequest request = new ListBucketsRequest();
            request.setBucketType(BucketTypeEnum.PFS);
            List<ObsBucket> buckets = obsClient.listBuckets(request);
            for (ObsBucket bucket : buckets) {
                System.out.println("BucketName:" + bucket.getBucketName());
                System.out.println("CreationDate:" + bucket.getCreationDate());
                System.out.println("Location:" + bucket.getLocation());
            }
            System.out.println("listBuckets successfully");
        } catch (ObsException e) {
            System.out.println("listBuckets failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
        }
    }
}
```

```
        System.out.println("Host ID:" + e.getErrorHostId());
        e.printStackTrace();
    } catch (Exception e) {
        System.out.println("listBuckets failed");
        // 其他异常信息打印
        e.printStackTrace();
    }
}
```

## 相关链接

- 关于并行文件系统的更多说明，请参见[什么是并行文件系统](#)。
- 更多关于列举并行文件系统的示例代码，请参见[Github示例](#)。
- 列举并行文件系统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](#)。
- 列举并行文件系统常见问题请参见[并行文件系统常见问题](#)。

## 8.3 列举并行文件系统内对象(Java SDK)

### 功能说明

您可以通过ObsClient.listObjects列举出指定并行文件系统里的对象。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 您必须是并行文件系统拥有者或拥有列举并行文件系统内对象的权限，才能列举并行文件系统内对象。建议使用IAM或策略进行授权，如果使用IAM则需授予obs:bucket:ListBucket权限，如果使用策略则需授予ListBucket权限。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以及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(final [ListObjectsRequest request](#))

## 请求参数说明

表 8-28 请求参数列表

参数名称	参数类型	是否必选	描述
request	ListObjectsRequest	必选	<b>参数解释:</b> 获取并行文件系统内对象列表请求参数, 详见 <a href="#">ListObjectsRequest</a> 。

表 8-29 ListObjectsRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释:</b> 桶名。 <b>约束限制:</b> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下:                             <ul style="list-style-type: none"> <li>3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻(如: “my..bucket”)。</li> <li>禁止“.”和“-”相邻(如: “my-.bucket”和“my-bucket”)。</li> </ul> </li> <li>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。</li> </ul> <b>默认取值:</b> 无
prefix	String	可选	<b>参数解释:</b> 限定返回的对象名必须带有prefix前缀。 <b>约束限制:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
marker	String	可选	<b>参数解释:</b> 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。 <b>约束限制:</b> 仅用于非多版本列举。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无
maxKeys	int	可选	<b>参数解释:</b> 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。 <b>取值范围:</b> 1~1000, 当超出范围时, 按照默认的1000进行处理。 <b>默认取值:</b> 1000

参数名称	参数类型	是否必选	描述
delimiter	String	可选	<p><b>参数解释:</b>                      将对象名进行分组的分隔符。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 commonPrefix; 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 commonPrefix, 而 bbcde 会被单独分成一组, 形成一条前缀为 bbcd 的 commonPrefix。</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><b>取值范围:</b>                      长度大于 0 且不超过 1024 的字符串。</p> <p><b>默认取值:</b>                      无</p>
encodingType	String	可选	<p><b>参数解释:</b>                      对响应中的部分元素进行指定类型的编码。如果 delimiter、marker、prefix、nextMarker 和 objectKey 包含 xml 1.0 标准不支持的控制字符, 可通过设置 encodingType 对响应中的 delimiter、marker、prefix (包括 commonPrefixes 中的 prefix)、nextMarker 和 objectKey 进行编码。</p> <p><b>取值范围:</b>                      可选值为 url。</p> <p><b>默认取值:</b>                      无, 不设置则不编码。</p>

## 返回结果说明

ObsClient.listObjects返回结果如下:



表 8-30 ObjectListing

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 并行文件系统的并行文件系统名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>并行文件系统的名字需全局唯一，不能与已有的任何并行文件系统名称重复，包括其他用户创建的并行文件系统。</li> <li>并行文件系统命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>同一用户在同一个区域多次创建同名并行文件系统不会报错，创建的并行文件系统属性以第一次请求为准。</li> </ul> <p><b>默认取值:</b> 无</p>
prefix	String	<p><b>参数解释:</b> 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，将返回并行文件系统的所有对象。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
objectSummaries	List<ObsObject>	<p><b>参数解释:</b> 并行文件系统内对象列表，详见<a href="#">ObsObject</a>。</p>

参数名称	参数类型	描述
commonPrefixes	List<String>	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
truncated	boolean	<p><b>参数解释:</b> 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><b>取值范围:</b> true: 表示没有返回全部结果。 false: 表示已返回了全部结果。</p> <p><b>默认取值:</b> 无</p>
marker	String	<p><b>参数解释:</b> 列举对象的起始位置, 与请求中的该参数对应。列举并行文件系统内对象列表时, 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><b>约束限制:</b> 仅用于非多版本列举。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
maxKeys	int	<p><b>参数解释:</b> 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> <p><b>取值范围:</b> 1~1000, 当超出范围时, 按照默认的1000进行处理。</p> <p><b>默认取值:</b> 1000</p>

参数名称	参数类型	描述
delimiter	String	<p><b>参数解释:</b> 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，并行文件系统中存在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> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
nextMarker	String	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
location	String	<p><b>参数解释:</b> 并行文件系统所在的区域。</p> <p><b>约束限制:</b> 该参数定义了并行文件系统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><b>取值范围:</b> 当前有效的OBS区域和终端节点的更多信息, 请参考<a href="#">地区和终端节点</a>。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><b>默认取值:</b> 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>
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表, 由多个元组构成。元组中String代表响应消息头的名称, Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

表 8-31 ObsObject

参数名称	参数类型	描述
objectKey	String	<p><b>参数解释:</b> 对象名。对象名是对象在存储并行文件系统中的唯一标识。对象名是对象在并行文件系统中的完整路径，路径中不包含并行文件系统名。 例如，您对象的访问地址为 examplebucket.obs.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
bucketName	String	<p><b>参数解释:</b> 并行文件系统名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>并行文件系统的名字需全局唯一，不能与已有的任何并行文件系统名称重复，包括其他用户创建的并行文件系统。</li><li>并行文件系统命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>同一用户在同一个区域多次创建同名并行文件系统不会报错，创建的并行文件系统属性以第一次请求为准。</li></ul> <p><b>默认取值:</b> 无</p>
objectContent	InputStream	<p><b>参数解释:</b> 获取对象的数据流。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
owner	<b>Owner</b>	<p><b>参数解释:</b> 对象的所有者，包含对象拥有者DomainId和对象拥有者名称，详见<b>Owner</b>。</p> <p><b>默认取值:</b> 无。</p>
metadata	<b>ObjectMetadata</b>	<p><b>参数解释:</b> 对象的元数据信息。</p> <p><b>取值范围:</b> 对象的元数据信息，详见<b>ObjectMetadata</b>。</p> <p><b>默认取值:</b> 无</p>

表 8-32 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID，即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a></p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a></p> <p><b>默认取值:</b> 无</p>

表 8-33 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 8-34 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 单次上传对象大小范围是[0, 5GB]。</li> <li>• 如果需要上传超过5GB的大文件，需要通过<a href="#">多段操作</a>来分段上传。</li> </ul> <p><b>默认取值:</b> 如果不设置，则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentType	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>
expires	String	可选	<p><b>参数解释:</b> 指对象在网页中的缓存过期时间。</p> <p><b>约束限制:</b> 日期格式为GMT的格式。</p> <p><b>默认取值:</b> 无</p>
contentMd5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>对象数据的MD5值必须经过Base64编码。</li> <li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li> </ul> <p><b>取值范围:</b> 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：n58IG6hfM7vqI4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
storageClass	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域设置对象的存储类别。如果未设置此头域，则以并行文件系统的默认存储类别作为对象的存储类别。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 当并行文件系统设置了Website配置，可以将获取这个对象的请求重定向到并行文件系统中另一个对象或一个外部的URL，该参数指明对象的重定向地址。</p> <p>例如，重定向请求到并行文件系统中另一对象： WebsiteRedirectLocation:/anotherPage.html 或重定向请求到一个外部URL： WebsiteRedirectLocation:http://www.example.com/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>必须以“/”、“http://”或“https://”开头，长度不超过2KB。</li> <li>OBS仅支持为并行文件系统根目录下的对象设置重定向，不支持为并行文件系统中文件夹下的对象设置重定向。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度，单位：字节。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称, Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取, 参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据, 总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时, 对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于列举examplebucket并行文件系统目录下的内容。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.*;
import java.io.*;

public class ListPosixObjects{
    // 列举对象返回的最大数量
    private static int maxKey = 1000;
    private static String bucketName = "examplebucket";
    private static ObsClient obsClient;
    public static void main(String[] args) throws IOException, InterruptedException {
        // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险, 建议在配置文件或者环境变量中密文存放, 使用时解密, 确保安全; 本示例以ak和sk保存在环境变量中为例, 运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca_01_0003.html
        String ak = System.getenv("OBS_ACCESS_KEY_ID");
        String sk = System.getenv("OBS_SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken, 也可以使用其他外部引入方式传入。
    }
}
```

```
String securityToken = System.getenv("OBS_SECURITY_TOKEN");
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
String endPoint = System.getenv("OBS_ENDPOINT");
// 创建ObsClient实例
try {
    obsClient = new ObsClient(ak, sk, endPoint);
    // 列举并行文件系统内的对象
    ListObjectsRequest request = new ListObjectsRequest(bucketName);
    // 列举对象返回的最大数量
    request.setMaxKeys(maxKey);
    // 建议设置[delimiter="/"], 只列举当前目录下的内容，不列举子目录
    request.setDelimiter("/");
    listObjects(request);
} catch (ObsException e) {
    // 请求失败,打印http状态码
    System.out.println("HTTP Code: " + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message: " + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
} catch (Exception e) {
    // 其他异常信息打印
    e.printStackTrace();
}
}

static void listObjectsByPrefix(ObjectListing result) throws ObsException {
    for (String prefix : result.getCommonPrefixes()) {
        System.out.println("Objects in folder [" + prefix + "]:");
        ListObjectsRequest request = new ListObjectsRequest(bucketName);
        // 列举对象的最大数目，取值范围为1~1000，当超出范围时，按照默认的1000进行处理
        request.setMaxKeys(maxKey);
        // 建议设置[delimiter="/"], 只列举当前目录下的内容，不列举子目录
        request.setDelimiter("/");
        // 按设置的对象前缀返回结果
        request.setPrefix(prefix);
        listObjects(request);
    }
}

static void listObjects(ListObjectsRequest request) {
    ObjectListing result;
    do {
        result = obsClient.listObjects(request);
        for (ObsObject obsObject : result.getObjects()) {
            // 打印列举的对象名
            System.out.println("\t" + obsObject.getObjectKey());
            // 打印该对象的所有者
            System.out.println("\t" + obsObject.getOwner());
        }
        // 列举对象的起始位置
        request.setMarker(result.getNextMarker());
        // 递归列举文件夹
        listObjectsByPrefix(result);
    } while (result.isTruncated());
}
}
```

## 相关链接

- 关于并行文件系统的更多说明，请参见[什么是并行文件系统](#)。
- 列举并行文件系统内对象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](#)。
- 列举并行文件系统内对象常见问题请参见[并行文件系统常见问题](#)。

## 8.4 修改写对象(Java SDK)

### 功能说明

将指定并行文件系统内的一个对象从指定位置起修改为其他内容。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 您必须是并行文件系统所有者或拥有修改写对象的权限，才能修改写对象。建议使用IAM或策略进行授权，如果使用IAM则需授予obs:bucket:PutObject权限，如果使用策略则需授予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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 目前接口仅在并行文件系统支持，普通对象并行文件系统不支持，如何创建并行文件系统请参考[创建并行文件系统\(Java SDK\)](#)。

### 方法定义

obsClient.modifyObject([ModifyObjectRequest request](#))

### 请求参数说明

表 8-35 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">ModifyObjectRequest</a>	必选	<b>参数解释：</b> 修改写对象请求参数，详见 <a href="#">ModifyObjectRequest</a> 。

表 8-36 ModifyObjectRequest

参数名称	参数类型	是否可选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
position	long	必选	<p><b>参数解释:</b> 修改写操作的位置。</p> <p><b>取值范围:</b> 不小于0的长整型。单位：字节。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否可选	描述
input	InputStream	可选	<b>参数解释:</b> 待上传对象的数据流。 <b>默认取值:</b> 无
file	File	可选	<b>参数解释:</b> 待上传的本地文件。 <b>默认取值:</b> 无
encodeHeaders	boolean	可选	<b>参数解释:</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制，无法发送非ASCII码字符，SDK会在发送请求时对您头域中的 <b>中文汉字</b> 进行url编码，发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”，则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。 <b>取值范围:</b> true: 启用SDK编码。 false: 不启用SDK编码。 <b>默认取值:</b> true

## 返回结果说明

表 8-37 ModifyObjectResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于指定位置修改examplebucket并行文件系统中objectname对象的内容。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ModifyObjectRequest;
import com.obs.services.model.ModifyObjectResult;
import java.io.ByteArrayInputStream;
public class ModifyObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 第一次上传
            ModifyObjectRequest request = new ModifyObjectRequest();
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname");
            request.setPosition(0);
            request.setInput(new ByteArrayInputStream("HELLO OBS FIRST".getBytes()));
            ModifyObjectResult result = obsClient.modifyObject(request);
            // 第二次修改写
            request.setPosition(0);
            request.setInput(new ByteArrayInputStream("hello obs second".getBytes()));
            result = obsClient.modifyObject(request);
            System.out.println("modifyObject successfully");
            System.out.println("HTTP Code: " + result.getStatusCode());
        } catch (ObsException e) {
            System.out.println("modifyObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
        }
    }
}
```

```
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("modifyObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于修改写对象的API说明，请参见[修改写对象](#)。
- 更多关于修改写对象的示例代码，请参见[Github示例](#)。
- 修改写对象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](#)。
- 修改写对象常见问题请参见[我可以修改对象名称吗？](#)。

## 8.5 重命名对象(Java SDK)

### 功能说明

重命名对象操作是指将指定并行文件系统内的一个对象重命名为其他对象名。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 您必须是并行文件系统所有者或拥有重命名文件的权限，才能重命名文件。建议使用IAM或策略进行授权，如果使用IAM则需授予obs:bucket:PutObject权限，如果使用策略则需授予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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 目前接口仅在并行文件系统支持，普通对象并行文件系统不支持，如何创建并行文件系统请参考[创建并行文件系统\(Java SDK\)](#)。

### 方法定义

obsClient.renameObject([RenameObjectRequest request](#))



## 请求参数说明

表 8-38 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">RenameObjectRequest</a>	必选	<b>参数解释：</b> 重命名对象请求参数，详见 <a href="#">RenameObjectRequest</a> 。

表 8-39 RenameObjectRequest

参数名称	参数类型	是否可选	描述
bucketName	String	必选	<b>参数解释：</b> 桶名。 <b>约束限制：</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my.bucket”和“my.bucket”）。</li></ul></li><li>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。</li></ul> <b>默认取值：</b> 无

参数名称	参数类型	是否可选	描述
objectKey	String	必选	<p><b>参数解释:</b> 原对象名称。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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
newObjectKey	String	必选	<p><b>参数解释:</b> 修改后的目标对象名称。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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 8-40 RenameObjectResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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。</p> <p>完整的状态码列表请参见<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于将examplebucket并行文件系统中的folderName/originalObjectName对象重命名为newFolderName/newObjectName。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.RenameObjectRequest;
import com.obs.services.model.RenameObjectResult;
public class RenameObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 重命名对象
            RenameObjectRequest request = new RenameObjectRequest();
            request.setBucketName("examplebucket");
            // objectKey 为原对象的完整对象名
            request.setObjectKey("folderName/originalObjectName");
            // newObjectKey 为目标对象的完整对象名
            request.setNewObjectKey("newFolderName/newObjectName");
            RenameObjectResult result = obsClient.renameObject(request);
            System.out.println("renameObject successfully");
            System.out.println("HTTP Code:" + result.getStatusCode());
        } catch (ObsException e) {
            System.out.println("renameObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
        }
    }
}
```

```
e.printStackTrace();
} catch (Exception e) {
    System.out.println("renameObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于重命名对象的API说明，请参见[重命名对象](#)。
- 更多关于重命名对象的示例代码，请参见[Github示例](#)。
- 重命名对象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](#)。
- 重命名对象常见问题请参见[我可以修改对象名称吗？](#)。

## 8.6 截断对象(Java SDK)

### 功能说明

截断对象操作是指将指定并行文件系统内的一个对象截断到指定大小。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 您必须是并行文件系统所有者或拥有截断对象的权限，才能截断对象。建议使用IAM或策略进行授权，如果使用IAM则需授予obs:bucket:PutObject权限，如果使用策略则需授予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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 目前接口仅在并行文件系统支持，普通对象并行文件系统不支持，如何创建并行文件系统请参考[创建并行文件系统\(Java SDK\)](#)。

### 方法定义

obsClient.truncateObject([TruncateObjectRequest request](#))

## 请求参数说明

表 8-41 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">TruncateObjectRequest</a>	必选	<b>参数解释：</b> 截断对象请求参数，详见 <a href="#">TruncateObjectRequest</a> 。

表 8-42 TruncateObjectRequest

参数名称	参数类型	是否可选	描述
bucketName	String	必选	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值：</b> 无</p>

参数名称	参数类型	是否可选	描述
objectKey	String	必选	<p><b>参数解释:</b> 文件对象名。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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
newLength	long	必选	<p><b>参数解释:</b> 文件截断后的大小。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 8-43 TruncateObjectResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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。</p> <p>完整的状态码列表请参见<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于将examplebucket并行文件系统中的objectname对象截断到指定大小，大小指定为1024L。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.TruncateObjectRequest;
import com.obs.services.model.TruncateObjectResult;
public class TruncateObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 截断操作
            TruncateObjectRequest request = new TruncateObjectRequest();
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname");
            request.setNewLength(1024L);
            TruncateObjectResult result = obsClient.truncateObject(request);
            System.out.println("truncateObject successfully");
            System.out.println("HTTP Code: " + result.getStatusCode());
        } catch (ObsException e) {
            System.out.println("truncateObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("truncateObject failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于截断对象的API说明，请参见[截断对象](#)。
- 更多关于截断对象的示例代码，请参见[Github示例](#)。
- 截断对象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](#)。

## 8.7 设置对象 AccessLabel

### 功能说明

OBS支持为对象设置AccessLabel进行权限控制处理。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 建议使用IAM或策略进行授权，如果使用IAM则需授予obs:bucket:PutAccessLabel权限，如果使用策略则需授予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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 目前接口仅在并行文件系统支持，普通对象桶不支持，如何创建并行文件系统请参考[创建并行文件系统\(Java SDK\)](#)。
- 仅支持并行文件系统的目录级别的设置，并行文件系统级和文件级均不支持。
- 用户需要具有“PutAccessLabel”权限才能设置对象AccessLabel。

### 方法定义

obsClient.SetAccessLabel([SetAccessLabelRequest request](#))

### 请求参数说明

表 8-44 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">SetAccessLabelRequest</a>	必选	参数解释： 请求参数， <a href="#">SetAccessLabelRequest</a> 。



表 8-45 SetAccessLabelRequest

参数名称	参数类型	是否可选	描述
bucketName	String	必选	<p><b>参数解释:</b> 并行文件系统名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>并行文件系统的名字需全局唯一，不能与已有的任何并行文件系统名称重复，包括其他用户创建的并行文件系统。</li><li>并行文件系统命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>同一用户在同一个区域多次创建同名并行文件系统不会报错，创建的并行文件系统属性以第一次请求为准。</li></ul> <p><b>默认取值:</b> 无</p>
Dir	String	必选	<p><b>参数解释:</b> 目录名。目录名是并行文件系统中目录的路径，路径中不包含并行文件系统名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否可选	描述
RoleLabel	List<String>	必选	<p><b>参数解释:</b> 目录的accesslabel内容。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>仅包含0-9、a-z、A-Z、下划线的字符串,最大长度为52个字符。</li> <li>目录级别AccessLabel最大设置支持256条。</li> </ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 8-46 SetAccessLabelResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于设置并行文件系统examplebucket中目录名为Dirname的accesslabel。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.fs.accesslabel.*
public class SetAccessLabel{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint, 此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");
```

```
// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    //设置目录accesslabel
    SetAccessLabelRequest request = new SetAccessLabelRequest();
    request.setBucketName("examplebucket");
    request.setDir("Dirname");
    List<String> list = Arrays.asList("Role1","Role2",);
    request.setRoleLabel(list);
    SetAccessLabelRequest result = obsClient.SetAccessLabelRequest(request);
    System.out.println("SetAccessLabel successfully");
    System.out.println("HTTP Code: " + result.getStatusCode());
} catch (ObsException e) {
    System.out.println("SetAccessLabel failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("SetAccessLabel failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 更多示例代码，请参见[Github示例](#)。
- 设置目录accesslabel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](#)。

## 8.8 获取对象 AccessLabel

### 功能说明

OBS支持为对象设置AccessLabel进行权限控制处理，此接口提供返回对象设置的AccessLabel标签信息。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 接口约束

- 建议使用IAM或策略进行授权，如果使用IAM则需授予obs:bucket:PutAccessLabel权限，如果使用策略则需授予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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 目前接口仅在并行文件系统支持，普通对象桶不支持，如何创建并行文件系统请参考[创建并行文件系统\(Java SDK\)](#)。
- 仅支持并行文件系统的目录级别的设置，并行文件系统级和文件级均不支持。

## 方法定义

obsClient.GetAccessLabel([GetAccessLabelRequest request](#))

## 请求参数说明

表 8-47 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">GetAccessLabelRequest</a>	必选	<b>参数解释：</b> 请求参数，详见 <a href="#">GetAccessLabelRequest</a> 。

表 8-48 GetAccessLabelRequest

参数名称	参数类型	是否可选	描述
bucketName	String	必选	<p><b>参数解释:</b> 并行文件系统名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>并行文件系统的名字需全局唯一，不能与已有的任何并行文件系统名称重复，包括其他用户创建的并行文件系统。</li><li>并行文件系统命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>同一用户在同一个区域多次创建同名并行文件系统不会报错，创建的并行文件系统属性以第一次请求为准。</li></ul> <p><b>默认取值:</b> 无</p>
Dir	String	必选	<p><b>参数解释:</b> 目录名。对象名是对象在存储并行文件系统中的唯一标识。目录名是并行文件系统中目录的路径，路径中不包含并行文件系统名。</p> <p>例如，您对象的访问地址为 examplebucket.obs.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 8-49 GetAccessLabelResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
RoleLabel	List<String>	<p><b>参数解释:</b> 目录的accesslabel内容。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>仅包含0-9、a-z、A-Z、下划线的字符串，最大长度为52个字符。</li> <li>目录级别AccessLabel最大设置支持256条。</li> </ul> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于获取examplebucket并行文件系统中的目录Dirname的accesslabel

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.fs.accesslabel.*
public class GetAccessLabel{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);
```

```
try {
    //获取目录accesslabel
    GetAccessLabelRequest request = new GetAccessLabelRequest();
    request.setBucketName("examplebucket");
    request.setDir("Dirname");
    GetAccessLabelRequest result = obsClient.GetAccessLabelRequest(request);
    System.out.println("GetAccessLabel successfully");
    System.out.println("HTTP Code: " + result.getStatusCode());
} catch (ObsException e) {
    System.out.println("GetAccessLabel failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("GetAccessLabel failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 更多示例代码，请参见[Github示例](#)。
- 设置目录accesslabel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](#)。

## 8.9 删除对象 AccessLabel

### 功能说明

OBS支持为对象设置AccessLabel进行权限控制处理，此接口提供删除已设置在对象中的AccessLabel。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 建议使用IAM或策略进行授权，如果使用IAM则需授予obs:bucket:PutAccessLabel权限，如果使用策略则需授予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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 目前接口仅在并行文件系统支持，普通对象桶不支持，如何创建并行文件系统请参考[创建并行文件系统\(Java SDK\)](#)。

- 仅支持并行文件系统的目录级别的设置，并行文件系统级和文件级均不支持。

## 方法定义

obsClient.DeleteAccessLabel([DeleteAccessLabelRequest request](#))

## 请求参数说明

表 8-50 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">DeleteAccessLabelRequest</a>	必选	<b>参数解释：</b> 请求参数，详见 <a href="#">DeleteAccessLabelRequest</a> 。

表 8-51 DeleteAccessLabelRequest

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



参数名称	参数类型	是否可选	描述
Dir	String	必选	<p><b>参数解释:</b> 目录名。目录名是并行文件系统中目录的路径，路径中不包含并行文件系统名。 例如，您对象的访问地址为 examplebucket.obs.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 8-52 DeleteAccessLabelResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
RoleLabel	List<String>	<p><b>参数解释:</b> 目录的accesslabel内容。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>仅包含0-9、a-z、A-Z、下划线的字符串，最大长度为52个字符。</li><li>目录级别AccessLabel最大设置支持256条。</li></ul> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于删除examplebucket并行文件系统中的目录Dirname的accesslabel。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.fs.accesslabel.*
public class DeleteAccessLabel{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            //获取目录accesslabel
            DeleteAccessLabelRequest request = new DeleteAccessLabelRequest();
            request.setBucketName("examplebucket");
            request.setDir("Dirname");
            DeleteAccessLabelRequest result = obsClient.DeleteAccessLabelRequest(request);
            System.out.println("DeleteAccessLabel successfully");
            System.out.println("HTTP Code: " + result.getStatusCode());
        } catch (ObsException e) {
            System.out.println("DeleteAccessLabel failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("DeleteAccessLabel failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 更多示例代码，请参见[Github示例](#)。
- 设置目录accesslabel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 上传对象(Java SDK)

## 9.1 对象上传简介(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

在OBS中，用户操作的基本数据单元是对象。OBS Java 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：

- [流式上传](#)
- [文件上传](#)
- [分段上传](#)
- [追加上传](#)
- [断点续传上传](#)
- [基于表单上传](#)

SDK支持上传0KB~5GB的对象。流式上传、文件上传和追加上传的内容大小不能超过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。

## 9.2 流式上传(Java SDK)

### 功能说明

您可以将本地文件直接通过Internet上传至OBS指定的位置。待上传的文件可以是任何类型：文本文件、图片、视频等。

通过SDK的流式上传，可以上传小于5GB的文件。本章节介绍如何使用Java SDK流式上传对象。流式上传使用java.io.InputStream作为对象的数据源，您可以通过ObsClient.putObject上传您的数据流到OBS。

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 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以及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，需要通过[分段上传\(Java SDK\)](#)来分段上传。
- 推荐使用[文件上传\(Java SDK\)](#)的形式上传本地文件，而不是文件流形式。

## 方法定义

obsClient.putObject([PutObjectRequest request](#))

## 请求参数说明

表 9-1 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">PutObjectRequest</a>	必选	<b>参数解释：</b> 上传对象请求参数，详见 <a href="#">PutObjectRequest</a> 。

表 9-2 PutObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
metadata	<b>ObjectMetadata</b>	可选	<p><b>参数解释:</b> 对象元数据，详见<b>ObjectMetadata</b>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 创建对象时可指定的预定义访问策略，可选择的访问策略选项参见<a href="#">AccessControlList</a>，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> 无</p>
sseKmsHeader	<a href="#">SseKmsHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseKmsHeader</a>。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseCHeader</a></p> <p><b>默认取值:</b> 无</p>
input	java.io.InputStream	可选	<p><b>参数解释:</b> 待上传对象的数据流。</p> <p><b>默认取值:</b> 无</p>
file	java.io.File	可选	<p><b>参数解释:</b> 待上传对象的文件。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
extensionPermissionMap	Map<ExtensionObjectPermissionEnum, Set<String>>	可选	<p><b>参数解释:</b> 桶ACL的授权Map, 您可以为一个或多个账号授予桶权限。Map的ExtensionObjectPermissionEnum用于指定权限, Map的Set&lt;String&gt;用于说明该权限授予的账号ID列表, 即domain_id列表。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>授予权限的范围详见ExtensionObjectPermissionEnum。</li> <li>如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a></li> </ul> <p><b>默认取值:</b> 无</p>
expires	int	可选	<p><b>参数解释:</b> 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><b>约束说明:</b> 此字段对于每个对象支持上传时配置, 也支持后期通过修改元数据接口附带头域x-obs-expires修改。</p> <p><b>取值范围:</b> 大于0的整型数, 单位: 天。</p> <p><b>默认取值:</b> 无</p>
progressListener	ProgressListener	可选	<p><b>参数解释:</b> 上传进度, 详见ProgressListener。</p>

参数名称	参数类型	是否必选	描述
encodeHeaders	boolean	可选	<p><b>参数解释：</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制，无法发送非ASCII码字符，SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码，发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”，则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围：</b> true：启用SDK编码。 false：不启用SDK编码。</p> <p><b>默认取值：</b> true</p>

表 9-3 ACL 预定义访问策略

常量名	说明
AccessControlList.REST_CANNED_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。
AccessControlList.REST_CANNED_PUBLIC_READ	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。 设在对象上，所有人可以获取该对象内容和元数据。
AccessControlList.REST_CANNED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControlList.REST_CANNED_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，可以获取该桶内对象的内容和元数据。 不能应用于对象。



常量名	说明
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 9-4 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释：</b> 对象数据的长度。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"> <li>• 单次上传对象大小范围是[0, 5GB]。</li> <li>• 如果需要上传超过5GB的大文件，需要通过<a href="#">多段操作</a>来分段上传。</li> </ul> <p><b>默认取值：</b> 如果不设置，则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释：</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围：</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值：</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释：</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值：</b> 无</p>

参数名称	参数类型	是否必选	描述
contentDisposition	String	可选	<b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。 <b>默认取值:</b> 无
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无
contentType	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentMd5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <p><b>取值范围:</b> 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：n58lG6hfM7vql4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>
storageClass	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li><li>• 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。</li></ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 9-5 ExtensionObjectPermissionEnum

常量值	说明
GRANT_READ	授权指定租户有读对象和获取对象元数据的权限。
GRANT_READ_ACP	授权指定租户有获取对象ACL的权限。
GRANT_WRITE_ACP	授权指定租户有写对象ACL的权限。
GRANT_FULL_CONTROL	授权指定租户有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。

表 9-6 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	<b>Owner</b>	可选	<p><b>参数解释:</b> 桶所有者的信息，详见<a href="#">Owner</a>。</p>

参数名称	参数类型	是否必选	参数类型
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。</p> <p><b>默认取值:</b> false</p>
grants	Set< <a href="#">GrantAndPermission</a> >	可选	<p><b>参数解释:</b> 被授权用户相关信息, 详见<a href="#">GrantAndPermission</a>。</p>

表 9-7 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<p><b>参数解释:</b> 被授权用户或用户组, 详见<a href="#">GranteeInterface</a>。</p>
permission	<a href="#">Permission</a>	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 9-8 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 9-9 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-10 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 9-11 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-12 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。



常量名	原始值	说明
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

表 9-13 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释：</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围：</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard, AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值：</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释：</b> 加密算法。</p> <p><b>约束限制：</b> 只支持AES256。</p> <p><b>取值范围：</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值：</b> 无</p>

参数名称	参数类型	是否必选	描述
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过 Base64编码后得到, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 9-14 SseKmsHeader

参数名称	参数类型	是否必选	描述
encryption	<a href="#">ServerEncryption</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-KMS方式, 对象使用SSE-KMS方式加密。</p> <p><b>取值范围:</b> 可选值: kms, 即选择SSE-KMS方式加密对象, 详见<a href="#">ServerEncryption</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持kms。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
kmsKeyId	String	可选	<p><b>参数解释:</b> SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p><b>取值范围:</b> 有效值支持两种格式:</p> <ol style="list-style-type: none"> <li>regionID:domainID(账号ID):key/key_id</li> <li>key_id</li> </ol> <p>其中:</p> <ul style="list-style-type: none"> <li>regionID是使用密钥所属region的ID, 可在<a href="#">地区和终端节点</a>页面获取;</li> <li>domainID是使用密钥所属账号的账号ID, 获取方法参见<a href="#">如何获取账号ID和用户ID?</a>;</li> <li>key_id是从数据加密服务创建的密钥ID, 获取方法请参见<a href="#">查看密钥</a>。</li> </ul> <p><b>默认取值:</b></p> <ul style="list-style-type: none"> <li>如果用户没有提供该头域, 那么默认的主密钥将会被使用。</li> <li>如果默认主密钥不存在, 将默认创建并使用。</li> </ul>

表 9-15 ServerEncryption

常量值	原始值
OBS_KMS	kms

表 9-16 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 9-17 ServerAlgorithm

常量值	原始值
AES256	AES256

表 9-18 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释:</b> 获取上传进度, 详见 <a href="#">progressChanged</a> 。 <b>默认取值:</b> 无

表 9-19 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释:</b> 传输进度数据, 详见 <a href="#">ProgressStatus</a> 。 <b>默认取值:</b> 无

表 9-20 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 9-21 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。

常量名	原始值	说明
COLD	COLD	归档存储。

## 返回结果说明

表 9-22 PutObjectResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
storageClass	<a href="#">StorageClassEnum</a>	<b>参数解释:</b> 对象的存储类别。当对象存储类别是标准存储时，该值为空。 <b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无
versionId	String	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	描述
etag	String	<p><b>参数解释:</b> 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。上传的对象或拷贝操作创建的对象, 都有唯一的etag。</p> <p><b>约束限制:</b> 当对象是服务端加密的对象时, etag值不是对象的MD5值。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中, 对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

## 代码示例：上传字符串（byte 数组）

上传字符串“Hello OBS”，上传后的对象命名为objectname，对象存储在名为examplebucket的桶中。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import java.io.ByteArrayInputStream;
public class PutObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);
```

```
try {
    // 上传字符串 (byte数组)
    String content = "Hello OBS";
    obsClient.putObject("examplebucket", "objectname", new
ByteArrayInputStream(content.getBytes()));
    System.out.println("putObject successfully");
} catch (ObsException e) {
    System.out.println("putObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("putObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：上传网络流

上传http://www.a.com网络流，上传后的对象名为objectname，对象存储在名为examplebucket的桶中。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import java.io.InputStream;
import java.net.URL;
public class PutObject002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 上传网络流
            InputStream inputStream = new URL("http://www.a.com").openStream();
            obsClient.putObject("examplebucket", "objectname", inputStream);
            System.out.println("putObject successfully");
        } catch (ObsException e) {
            System.out.println("putObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
        }
    }
}
```



```
        System.out.println("Error Message:" + e.getMessage());
        // 请求失败,打印请求id
        System.out.println("Request ID:" + e.getErrorRequestId());
        System.out.println("Host ID:" + e.getErrorHostId());
        e.printStackTrace();
    } catch (Exception e) {
        System.out.println("putObject failed");
        // 其他异常信息打印
        e.printStackTrace();
    }
}
```

## 代码示例：上传文件流

以下代码将localfile文件流上传到bucketname中dir目录下的objectname对象中。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.PutObjectRequest;
import java.io.File;
import java.io.FileInputStream;
public class PutObject003 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 待上传的本地文件路径，需要指定到具体的文件名
            FileInputStream fis = new FileInputStream(new File("localfile"));
            obsClient.putObject("examplebucket", "dir/objectname", fis);
            // 待上传的本地文件路径，需要指定到具体的文件名
            FileInputStream fis2 = new FileInputStream(new File("localfile2"));
            PutObjectRequest request = new PutObjectRequest();
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname2");
            request.setInput(fis2);
            obsClient.putObject(request);
            System.out.println("putObject successfully");
        } catch (ObsException e) {
            System.out.println("putObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
```

```
System.out.println("putObject failed");  
// 其他异常信息打印  
e.printStackTrace();  
}  
}  
}
```

## 相关链接

- 关于上传对象-PUT上传的API说明，请参见[PUT上传](#)。
- 更多关于上传对象的示例代码，请参见[Github示例](#)。
- 上传对象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](#)。
- 上传对象常见问题请参见[上传对象失败](#)。

## 9.3 文件上传(Java SDK)

### 功能说明

您可以将本地文件直接通过Internet上传至OBS指定的桶中。待上传的文件可以是任何类型。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及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，需要通过[分段上传](#)来分段上传。

### 方法定义

obsClient.putObject([PutObjectRequest request](#))

## 请求参数说明

表 9-23 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">PutObjectRequest</a>	必选	<b>参数解释：</b> 上传对象请求参数，详见 <a href="#">PutObjectRequest</a> 。

表 9-24 PutObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释：</b> 桶名。 <b>约束限制：</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <b>默认取值：</b> 无

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
metadata	<a href="#">ObjectMetadata</a>	可选	<p><b>参数解释:</b> 对象元数据，详见<a href="#">ObjectMetadata</a>。</p> <p><b>默认取值:</b> 无</p>
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 创建对象时可指定的预定义访问策略，您可选的访问策略选项参见<a href="#">AccessControlList</a>，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> 无</p>
sseKmsHeader	<a href="#">SseKmsHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseKmsHeader</a>。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseCHeader</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
input	java.io.InputStream	可选	<b>参数解释:</b> 待上传对象的数据流。 <b>默认取值:</b> 无
file	java.io.File	可选	<b>参数解释:</b> 待上传对象的文件。 <b>默认取值:</b> 无
extensionPermissionMap	Map<ExtensionObjectPermissionEnum, Set<String>>	可选	<b>参数解释:</b> 桶ACL的授权Map，您可以为一个或多个账号授予桶权限。Map的ExtensionObjectPermissionEnum用于指定权限，Map的Set<String>用于说明该权限授予的账号ID列表，即domain_id列表。 <b>取值范围:</b> <ul style="list-style-type: none"><li>授予权限的范围详见ExtensionObjectPermissionEnum。</li><li>如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a></li></ul> <b>默认取值:</b> 无
expires	int	可选	<b>参数解释:</b> 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。 <b>约束限制:</b> 此字段对于每个对象支持上传时配置，也支持后期通过修改元数据接口附带头域x-obs-expires修改。 <b>取值范围:</b> 大于0的整型数，单位：天。 <b>默认取值:</b> 无
progressListener	ProgressListener	可选	<b>参数解释:</b> 上传进度，详见 <a href="#">ProgressListener</a> 。

参数名称	参数类型	是否必选	描述
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制，无法发送非ASCII码字符，SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码，发送编码后数据。如您设置的值content-disposition为“attachment; filename=“中文.txt””，则对象元数据中存储的信息为“attachment; filename=%E4%B8%AD%E6%96%87.txt”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围:</b> true: 启用SDK编码。 false: 不启用SDK编码。</p> <p><b>默认取值:</b> true</p>

表 9-25 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<p><b>参数解释:</b> 获取上传进度，详见<a href="#">progressChanged</a>。</p> <p><b>默认取值:</b> 无</p>

表 9-26 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<p><b>参数解释:</b> 传输进度数据，详见<a href="#">ProgressStatus</a>。</p> <p><b>默认取值:</b> 无</p>

表 9-27 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 9-28 ExtensionObjectPermissionEnum

常量值	说明
GRANT_READ	授权指定租户有读对象和获取对象元数据的权限。
GRANT_READ_ACP	授权指定租户有获取对象ACL的权限。
GRANT_WRITE_ACP	授权指定租户有写对象ACL的权限。
GRANT_FULL_CONTROL	授权指定租户有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。

表 9-29 SseKmsHeader

参数名称	参数类型	是否必选	描述
encryption	<a href="#">ServerEncryption</a>	必选	<p><b>参数解释：</b> 以SSE-KMS方式加密对象。</p> <p><b>参数解释：</b> 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p><b>取值范围：</b> 可选值：kms，即选择SSE-KMS方式加密对象，详见<a href="#">ServerEncryption</a>。</p> <p><b>默认取值：</b> 无</p>

参数名称	参数类型	是否必选	描述
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<b>参数解释:</b> 加密算法。 <b>约束限制:</b> 只支持kms。 <b>取值范围:</b> 详见 <a href="#">SSEAlgorithmEnum</a> 。 <b>默认取值:</b> 无
kmsKeyId	String	可选	<b>参数解释:</b> SSE-KMS加密方式下使用的KMS主密钥的ID值。 <b>取值范围:</b> 有效值支持两种格式: 1. regionID:domainID(账号ID):key/key_id 2. key_id 其中: <ul style="list-style-type: none"><li>• regionID是使用密钥所属region的ID, 可在<a href="#">地区和终端节点</a>页面获取;</li><li>• domainID是使用密钥所属账号的账号ID, 获取方法参见<a href="#">如何获取账号ID和用户ID?</a>;</li><li>• key_id是从数据加密服务创建的密钥ID, 获取方法请参见<a href="#">查看密钥</a>。</li></ul> <b>默认取值:</b> <ul style="list-style-type: none"><li>• 如果用户没有提供该头域, 那么默认的主密钥将会被使用。</li><li>• 如果默认主密钥不存在, 将默认创建并使用。</li></ul>



表 9-30 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard, AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥，byte[]形式，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥，由原始密钥经过Base64编码后得到，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 9-31 ServerEncryption

常量值	原始值
OBS_KMS	kms

表 9-32 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 9-33 ServerAlgorithm

常量值	原始值
AES256	AES256

表 9-34 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	<b>Owner</b>	可选	<b>参数解释:</b> 桶所有者的信息, 详见 <b>Owner</b> 。
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false
grants	Set< <b>GrantAndPermission</b> >	可选	<b>参数解释:</b> 被授权用户相关信息, 详见 <b>GrantAndPermission</b> 。

表 9-35 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-36 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组, 详见 <a href="#">GranteeInterface</a> 。
permission	<a href="#">Permission</a>	必选	<b>参数解释:</b> 用户或用户组被授予的权限。 <b>取值范围:</b> 详见 <a href="#">Permission</a> 。 <b>默认取值:</b> 无
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false

表 9-37 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。
PERMISSION_WRITE_ACP	WRITE_ACP	修改ACL配置的权限。 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。
PERMISSION_FULL_CONTROL	FULL_CONTROL	完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。 如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。 如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。

表 9-38 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 9-39 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-40 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 9-41 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 单次上传对象大小范围是[0, 5GB]。</li> <li>• 如果需要上传超过5GB的大文件, 需要通过<a href="#">多段操作</a>来分段上传。</li> </ul> <p><b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型 ( MIME类型 )。 contentType ( 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><b>取值范围:</b> 常见的contentType ( MIME ) 列表参见<a href="#">如何理解Content-Type ( MIME ) ? (Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type, SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type ( 如.xml判断为application/xml文件; .html判断为text/html文件 )。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象, 当下载对象或者访问对象时, 以默认文件名命名的文件将直接在浏览器上显示, 或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无
contentType	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无
contentMd5	String	可选	<b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。 <b>约束限制:</b> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <b>取值范围:</b> 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：n58IG6hfM7vqI4K0vnWpog== <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
storageClass	StorageClassEnum	可选	<b>参数解释:</b> 对象的存储类别。创建对象时,可以加上此头域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。 <b>取值范围:</b> 可选择的存储类别参见StorageClassEnum。 <b>默认取值:</b> 无
websiteRedirectLocation	String	可选	<b>参数解释:</b> 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,该参数指明对象的重定向地址。 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/ <b>约束限制:</b> <ul style="list-style-type: none"><li>必须以“/”、“http://”或“https://”开头,长度不超过2KB。</li><li>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。</li></ul> <b>默认取值:</b> 无
nextPosition	long	可选	<b>参数解释:</b> 下次追加上传的位置。 <b>取值范围:</b> 0~对象长度,单位:字节。 <b>默认取值:</b> 无
appendable	boolean	可选	<b>参数解释:</b> 对象是否为可追加上传的对象。 <b>取值范围:</b> true: 是,可追加上传。 false: 否,不可追加上传。 <b>默认取值:</b> 无



参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 9-42 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 9-43 ACL 预定义访问策略

常量名	说明
AccessControlList.REST_CANNED_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。
AccessControlList.REST_CANNED_PUBLIC_READ	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。 设在对象上，所有人可以获取该对象内容和元数据。

常量名	说明
AccessControllist.REST_CAN NED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControllist.REST_CAN NED_PUBLIC_READ_DELIVER ED	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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControllist.REST_CAN NED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

## 返回结果说明

表 9-44 PutObjectResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释：</b> HTTP状态码。 <b>取值范围：</b> 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 <a href="#">状态码</a> 。 <b>默认取值：</b> 无
responseHeader s	Map<String, Object>	<b>参数解释：</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值：</b> 无

参数名称	参数类型	描述
storageClass	<a href="#">StorageClassEnum</a>	<b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。 <b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无
versionId	String	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
etag	String	<b>参数解释:</b> 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。上传的对象或拷贝操作创建的对象，都有唯一的etag。 <b>约束限制:</b> 当对象是服务端加密的对象时，etag值不是对象的MD5值。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	描述
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>

## 代码示例

文件上传使用本地文件作为对象的数据源。以下代码展示了将本地文件localfile上传到examplebucket桶中。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.PutObjectRequest;
import java.io.File;
public class PutObject004 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
    }
}
```

```
// 您可以登录访问管理控制台获取访问密钥AK/SK
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
// 【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/SecurityToken，也可以使用其他外部引入方式传入。
// String securityToken = System.getenv("SECURITY_TOKEN");
// endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk,endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 文件上传
    // localfile 为待上传的本地文件路径，需要指定到具体的文件名
    PutObjectRequest request = new PutObjectRequest();
    request.setBucketName("examplebucket");
    request.setObjectKey("objectkey");
    request.setFile(new File("localfile"));
    obsClient.putObject(request);
    System.out.println("putObject successfully");
} catch (ObsException e) {
    System.out.println("putObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("putObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于上传对象-POST上传的API说明，请参见[POST上传](#)。
- 更多关于上传对象的示例代码，请参见[Github示例](#)。
- 上传对象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](#)。
- 上传对象常见问题请参见[上传对象失败](#)。

## 9.4 获取上传进度(Java SDK)

### 功能说明

获取指定对象的上传进度。

您可以通过PutObjectRequest.setProgressListener设置数据传输接口来获取上传的进度。

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 接口约束

- 您必须拥有obs:object:PutObject权限，才能获取获取上传进度。相关授权操作可参见[典型权限场景配置案例](#)。
- OBS支持的region以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 支持获取上传进度的接口包括：流式上传、文件上传、上传段、追加上传和断点续传上传。
- 如果ProgressStatus.getTransferPercentage()返回-1，表明是流式上传，此时必须设置对象属性中的对象长度（Content-Length）。

## 方法定义

PutObjectRequest.setProgressListener([ProgressListener progressListener](#))

## 参数说明

表 9-45 请求参数列表

参数名称	参数类型	是否必选	描述
progressListener	<a href="#">ProgressListener</a>	必选	<b>参数解释：</b> 设置数据传输监听器，用于获取上传进度，详见 <a href="#">ProgressListener</a> 。

表 9-46 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释：</b> 获取上传进度，详见 <a href="#">progressChanged</a> 。

表 9-47 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	ProgressStatus	必选	参数解释： 传输进度数据，详见 ProgressStatus。

表 9-48 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

## 代码示例

本示例用于获取将exampleLocalFilePath本地文件上传到examplebucket桶中objectname对象的进度。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ProgressListener;
import com.obs.services.model.ProgressStatus;
import com.obs.services.model.PutObjectRequest;
import java.io.File;
public class PutObject005 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
```

```
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 文件上传
    PutObjectRequest request = new PutObjectRequest("examplebucket", "exampleobject");
    request.setFile(new File("exampleLocalFilePath"));
    request.setProgressListener(
        new ProgressListener() {
            @Override
            public void progressChanged(ProgressStatus status) {
                // 获取上传平均速率
                System.out.println("AverageSpeed:" + status.getAverageSpeed());
                // 获取上传进度百分比
                System.out.println("TransferPercentage:" + status.getTransferPercentage());
            }
        }
    );
    // 每上传1MB数据反馈上传进度
    request.setProgressInterval(1024 * 1024L);
    obsClient.putObject(request);
    System.out.println("putObject successfully");
} catch (ObsException e) {
    System.out.println("putObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("putObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 更多关于上传对象的示例代码，请参见[Github示例](#)。
- 上传对象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](#)。
- 上传对象常见问题请参见[上传对象失败](#)。

## 9.5 创建文件夹(Java SDK)

### 功能说明

您可以在已创建的桶中新建一个文件夹，从而更方便的对存储在OBS中的数据进行分类管理。

OBS本身是没有文件夹的概念的，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。



### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 接口约束

- 您必须是桶拥有者或拥有上传对象的权限，才能创建文件夹。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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 多级文件夹创建最后一级即可，比如src1/src2/src3/，创建src1/src2/src3/即可，无需创建src1/、src1/src2/。

## 方法定义

obsClient.putObject([PutObjectRequest request](#))

## 请求参数说明

表 9-49 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">PutObjectRequest</a>	必选	<b>参数解释：</b> 上传对象请求参数，详见 <a href="#">PutObjectRequest</a> 。

表 9-50 PutObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
metadata	<b>ObjectMetadata</b>	可选	<p><b>参数解释:</b> 对象元数据，详见<b>ObjectMetadata</b>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 创建对象时可指定的预定义访问策略，可选择的访问策略选项参见<a href="#">AccessControlList</a>，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> 无</p>
sseKmsHeader	<a href="#">SseKmsHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseKmsHeader</a>。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseCHeader</a>。</p> <p><b>默认取值:</b> 无</p>
input	java.io.InputStream	可选	<p><b>参数解释:</b> 待上传对象的数据流。</p> <p><b>默认取值:</b> 无</p>
file	java.io.File	可选	<p><b>参数解释:</b> 待上传对象的文件。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
extensionPermissionMap	Map<ExtensionObjectPermissionEnum, Set<String>>	可选	<p><b>参数解释:</b> 桶ACL的授权Map，您可以为一个或多个账号授予桶权限。Map的ExtensionObjectPermissionEnum用于指定权限，Map的Set&lt;String&gt;用于说明该权限授予的账号ID列表，即domain_id列表。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>授予权限的范围详见ExtensionObjectPermissionEnum。</li> <li>如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</li> </ul> <p><b>默认取值:</b> 无</p>
expires	int	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 此字段对于每个对象支持上传时配置，也支持后期通过修改元数据接口附带头域x-obs-expires修改。</p> <p><b>取值范围:</b> 大于0的整型数，单位：天。</p> <p><b>默认取值:</b> 无</p>
progressListener	ProgressListener	可选	<p><b>参数解释:</b> 上传进度，详见ProgressListener。</p>

参数名称	参数类型	是否必选	描述
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制,无法发送非ASCII码字符,SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码,发送编码后数据。如您设置的值content-disposition为“attachment; filename=“中文.txt””,则对象元数据中存储的信息为“attachment; filename=%E4%B8%AD%E6%96%87.txt”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围:</b> true: 启用SDK编码。 false: 不启用SDK编码。</p> <p><b>默认取值:</b> true</p>

表 9-51 ACL 预定义访问策略

常量名	说明
AccessControlList.REST_CANNED_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。
AccessControlList.REST_CANNED_PUBLIC_READ	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。 设在对象上,所有人可以获取该对象内容和元数据。
AccessControlList.REST_CANNED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControlList.REST_CANNED_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,可以获取该桶内对象的内容和元数据。 不能应用于对象。

常量名	说明
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 9-52 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释：</b> 获取上传进度，详见 <a href="#">progressChanged</a> 。 <b>默认取值：</b> 无

表 9-53 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释：</b> 传输进度数据，详见 <a href="#">ProgressStatus</a> 。 <b>默认取值：</b> 无

表 9-54 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。

方法名称	返回值类型	说明
getTransferredBytes() ( )	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 9-55 ExtensionObjectPermissionEnum

常量值	说明
GRANT_READ	授权指定租户有读对象和获取对象元数据的权限。
GRANT_READ_ACP	授权指定租户有获取对象ACL的权限。
GRANT_WRITE_ACP	授权指定租户有写对象ACL的权限。
GRANT_FULL_CONTROL	授权指定租户有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。

表 9-56 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard，AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过Base64编码后得到, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 9-57 SseKmsHeader

参数名称	参数类型	是否必选	描述
encryption	<a href="#">ServerEncryption</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p><b>取值范围:</b> 可选值: kms, 即选择SSE-KMS方式加密对象, 详见<a href="#">ServerEncryption</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持kms。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
kmsKeyId	String	可选	<p><b>参数解释:</b> SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p><b>取值范围:</b> 有效值支持两种格式:</p> <ol style="list-style-type: none"> <li>regionID:domainID(账号ID):key/key_id</li> <li>key_id</li> </ol> <p>其中:</p> <ul style="list-style-type: none"> <li>regionID是使用密钥所属region的ID,可在<a href="#">地区和终端节点</a>页面获取;</li> <li>domainID是使用密钥所属账号的账号ID,获取方法参见<a href="#">如何获取账号ID和用户ID?</a>;</li> <li>key_id是从数据加密服务创建的密钥ID,获取方法请参见<a href="#">查看密钥</a>。</li> </ul> <p><b>默认取值:</b></p> <ul style="list-style-type: none"> <li>如果用户没有提供该头域,那么默认的主密钥将会被使用。</li> <li>如果默认主密钥不存在,将默认创建并使用。</li> </ul>

表 9-58 ServerAlgorithm

常量值	原始值
AES256	AES256

表 9-59 ServerEncryption

常量值	原始值
OBS_KMS	kms

表 9-60 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 9-61 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 9-62 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 单次上传对象大小范围是[0, 5GB]。</li> <li>• 如果需要上传超过5GB的大文件，需要通过<a href="#">多段操作</a>来分段上传。</li> </ul> <p><b>默认取值:</b> 如果不设置，则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
contentDisposition	String	可选	<b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。 <b>默认取值:</b> 无
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无
contentType	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentMd5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <p><b>取值范围:</b> 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><b>默认取值:</b> 无</p>
storageClass	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 9-63 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	<b>Owner</b>	可选	<p><b>参数解释:</b> 桶所有者的信息，详见<a href="#">Owner</a>。</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>
grants	Set< <a href="#">GrantAndPermission</a> >	可选	<p><b>参数解释:</b> 被授权用户相关信息，详见<a href="#">GrantAndPermission</a>。</p>

表 9-64 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 9-65 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<p><b>参数解释:</b> 被授权用户或用户组, 详见<a href="#">GranteeInterface</a>。</p>
permission	<a href="#">Permission</a>	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 9-66 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。
PERMISSION_WRITE_ACP	WRITE_ACP	修改ACL配置的权限。 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。
PERMISSION_FULL_CONTROL	FULL_CONTROL	完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。 如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。 如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。



表 9-67 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 9-68 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-69 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

## 返回结果说明

表 9-70 PutObjectResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
storageClass	<a href="#">StorageClassEnum</a>	<b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。 <b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无
versionId	String	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	描述
etag	String	<p><b>参数解释:</b> 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。上传的对象或拷贝操作创建的对象, 都有唯一的etag。</p> <p><b>约束限制:</b> 当对象是服务端加密的对象时, etag值不是对象的MD5值。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中, 对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值：</b> 无</p>

## 代码示例

本示例用于在examplebucket桶中利用上传对象操作创建parent\_directory/ 文件夹。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import java.io.ByteArrayInputStream;
public class PutObject006 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
```

```
// 创建文件夹
final String keySuffixWithSlash = "parent_directory/";
obsClient.putObject("examplebucket", keySuffixWithSlash, new ByteArrayInputStream(new
byte[0]));
// 在文件夹下创建对象
obsClient.putObject(
    "examplebucket", keySuffixWithSlash + "objectname", new ByteArrayInputStream("Hello
OBS".getBytes()));
System.out.println("putObject successfully");
} catch (ObsException e) {
    System.out.println("putObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("putObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 通过API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。
- 更多关于创建文件夹的示例代码，请参见[Github示例](#)。
- 创建文件夹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](#)。

## 9.6 设置对象元数据(Java SDK)

### 功能说明

您可以在上传对象时设置对象元数据。对象元数据包含对象长度、对象MIME类型、对象MD5值（用于校验）、对象存储类别、对象自定义元数据。对象元数据可以在多种上传方式下（流式上传、文件上传、分段上传），或复制对象时进行设置。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 您必须是桶拥有者或拥有上传对象的权限，才能上传时设置对象元数据。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:object: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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。

- 一个对象可以有多个元数据，总大小不能超过8KB。
- 当前元数据名称不支持非ASCII码字符，元数据值包含非ASCII码字符时需进行Base64编码。

## 方法定义

obsClient.putObject([PutObjectRequest request](#))

## 请求参数说明

表 9-71 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">PutObjectRequest</a>	必选	<b>参数解释：</b> 上传对象请求参数，详见 <a href="#">PutObjectRequest</a> 。

表 9-72 PutObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释：</b> 桶名。 <b>约束限制：</b> <ul style="list-style-type: none"><li>• 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。</li><li>• 桶命名规则如下：<ul style="list-style-type: none"><li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>- 禁止使用IP地址。</li><li>- 禁止以“-”或“.”开头及结尾。</li><li>- 禁止两个“.”相邻（如：“my..bucket”）。</li><li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>• 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。</li></ul> <b>默认取值：</b> 无

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
metadata	<a href="#">ObjectMetadata</a>	可选	<p><b>参数解释:</b> 对象元数据，详见<a href="#">ObjectMetadata</a>。</p> <p><b>默认取值:</b> 无</p>
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 创建对象时可指定的预定义访问策略，可选择的访问策略选项参见<a href="#">AccessControlList</a>，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> 无</p>
sseKmsHeader	<a href="#">SseKmsHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseKmsHeader</a>。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseCHeader</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
input	java.io.InputStream	可选	<b>参数解释:</b> 待上传对象的数据流。 <b>默认取值:</b> 无
file	java.io.File	可选	<b>参数解释:</b> 待上传对象的文件。 <b>默认取值:</b> 无
extensionPermissionMap	Map<ExtensionObjectPermissionEnum, Set<String>>	可选	<b>参数解释:</b> 桶ACL的授权Map，您可以为一个或多个账号授予桶权限。Map的ExtensionObjectPermissionEnum用于指定权限，Map的Set<String>用于说明该权限授予的账号ID列表，即domain_id列表。 <b>取值范围:</b> <ul style="list-style-type: none"> <li>授予权限的范围详见ExtensionObjectPermissionEnum。</li> <li>如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</li> </ul> <b>默认取值:</b> 无
expires	int	可选	<b>参数解释:</b> 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。 <b>约束限制:</b> 此字段对于每个对象支持上传时配置，也支持后期通过修改元数据接口附带头域x-obs-expires修改。 <b>取值范围:</b> 大于0的整型数，单位：天。 <b>默认取值:</b> 无
progressListener	ProgressListener	可选	<b>参数解释:</b> 上传进度，详见ProgressListener。



参数名称	参数类型	是否必选	描述
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制,无法发送非ASCII码字符,SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码,发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”,则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围:</b> true: 启用SDK编码。 false: 不启用SDK编码。</p> <p><b>默认取值:</b> true</p>

表 9-73 ACL 预定义访问策略

常量名	说明
AccessControllist.REST_CANNED_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。
AccessControllist.REST_CANNED_PUBLIC_READ	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。 设在对象上,所有人可以获取该对象内容和元数据。
AccessControllist.REST_CANNED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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	说明
AccessControlList.REST_CANNED_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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 9-74 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释：</b> 获取上传进度，详见 <a href="#">progressChanged</a> 。 <b>默认取值：</b> 无

表 9-75 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释：</b> 传输进度数据，详见 <a href="#">ProgressStatus</a> 。 <b>默认取值：</b> 无

表 9-76 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。

方法名称	返回值类型	说明
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 9-77 ExtensionObjectPermissionEnum

常量值	说明
GRANT_READ	授权指定租户有读对象和获取对象元数据的权限。
GRANT_READ_ACP	授权指定租户有获取对象ACL的权限。
GRANT_WRITE_ACP	授权指定租户有写对象ACL的权限。
GRANT_FULL_CONTROL	授权指定租户有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。

表 9-78 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard，AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过Base64编码后得到, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 9-79 SseKmsHeader

参数名称	参数类型	是否必选	描述
encryption	<a href="#">ServerEncryption</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p><b>取值范围:</b> 可选值: kms, 即选择SSE-KMS方式加密对象, 详见<a href="#">ServerEncryption</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持kms。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
kmsKeyId	String	可选	<p><b>参数解释:</b> SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p><b>取值范围:</b> 有效值支持两种格式:</p> <ol style="list-style-type: none"> <li>regionID:domainID(账号ID):key/key_id</li> <li>key_id</li> </ol> <p>其中:</p> <ul style="list-style-type: none"> <li>regionID是使用密钥所属region的ID, 可在<a href="#">地区和终端节点</a>页面获取;</li> <li>domainID是使用密钥所属账号的账号ID, 获取方法参见<a href="#">如何获取账号ID和用户ID?</a>;</li> <li>key_id是从数据加密服务创建的密钥ID, 获取方法请参见<a href="#">查看密钥</a>。</li> </ul> <p><b>默认取值:</b></p> <ul style="list-style-type: none"> <li>如果用户没有提供该头域, 那么默认的主密钥将会被使用。</li> <li>如果默认主密钥不存在, 将默认创建并使用。</li> </ul>

表 9-80 ServerAlgorithm

常量值	原始值
AES256	AES256

表 9-81 ServerEncryption

常量值	原始值
OBS_KMS	kms

表 9-82 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 9-83 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 9-84 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<b>参数解释:</b> 对象数据的长度。 <b>约束限制:</b> <ul style="list-style-type: none"><li>• 单次上传对象大小范围是[0, 5GB]。</li><li>• 如果需要上传超过5GB的大文件, 需要通过<a href="#">多段操作</a>来分段上传。</li></ul> <b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentTypeLanguage	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
expires	String	可选	<p><b>参数解释:</b> 指对象在网页中的缓存过期时间。</p> <p><b>约束限制:</b> 日期格式为GMT的格式。</p> <p><b>默认取值:</b> 无</p>
contentMd5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <p><b>取值范围:</b> 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：n58lG6hfM7vql4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>
storageClasses	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 9-85 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	Owner	可选	<p><b>参数解释:</b> 桶所有者的信息，详见<a href="#">Owner</a>。</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>
grants	Set<GrantAndPermission>	可选	<p><b>参数解释:</b> 被授权用户相关信息，详见<a href="#">GrantAndPermission</a>。</p>

表 9-86 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 9-87 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<p><b>参数解释:</b> 被授权用户或用户组, 详见<a href="#">GranteeInterface</a>。</p>
permission	<a href="#">Permission</a>	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 9-88 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。
PERMISSION_WRITE_ACP	WRITE_ACP	修改ACL配置的权限。 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。
PERMISSION_FULL_CONTROL	FULL_CONTROL	完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。 如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。 如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。

表 9-89 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 9-90 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-91 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

## 返回结果说明

表 9-92 PutObjectResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
storageClass	<a href="#">StorageClassEnum</a>	<b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。 <b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无
versionId	String	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	描述
etag	String	<p><b>参数解释:</b> 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。上传的对象或拷贝操作创建的对象, 都有唯一的etag。</p> <p><b>约束限制:</b> 当对象是服务端加密的对象时, etag值不是对象的MD5值。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 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.cn- north-4.myhuaweicloud.com/folder/test.txt 中, 对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

## 代码示例：设置对象长度

本示例用于设置对象长度。将本地文件localfile上传到桶examplebucket中，将此对象命名为objectname，同时将objectname对象长度设置为1024 \* 1024L。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObjectMetadata;
import java.io.File;
import java.io.IOException;
public class PutObject007 {
    public static void main(String[] args) throws IOException {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);
```



```
try {
    // 设置对象长度
    ObjectMetadata metadata = new ObjectMetadata();
    metadata.setContentLength(1024 * 1024L); // 1MB
    obsClient.putObject("examplebucket", "objectname", new File("localfile"), metadata);
    System.out.println("putObject successfully");
} catch (ObsException e) {
    System.out.println("putObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("putObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：设置对象 MIME 类型

本示例通过设置对象MIME类型为image/jpeg，将本地文件localimage.jpg上传到examplebucket桶下的objectname.jpg对象中。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObjectMetadata;
import java.io.File;
public class PutObject008 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 设置对象MIME类型
            // 上传图片
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.setContentType("image/jpeg");
            obsClient.putObject("examplebucket", "objectname.jpg", new File("localimage.jpg"), metadata);
            System.out.println("putObject successfully");
        } catch (ObsException e) {
            System.out.println("putObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
        }
    }
}
```

```
System.out.println("Error Code:" + e.getErrorCode());
// 请求失败,打印详细错误信息
System.out.println("Error Message:" + e.getErrorMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("putObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：设置对象 MD5 值

本示例用于通过ObsClient.base64Md5来直接计算Content-MD5头域，将本地文件localimage.jpg上传到examplebucket桶下的objectname对象中。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObjectMetadata;
import java.io.File;
import java.nio.file.Files;
import java.nio.file.Paths;
public class PutObject009 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 设置对象MD5值
            // 上传图片
            ObjectMetadata metadata = new ObjectMetadata();
            //your file's md5 which encoded by base64
            String base64Md5 = obsClient.base64Md5(Files.newInputStream(Paths.get("localimage.jpg")));
            metadata.setContentMd5(base64Md5);
            obsClient.putObject("examplebucket", "objectname", new File("localimage.jpg"), metadata);
            System.out.println("putObject successfully");
        } catch (ObsException e) {
            System.out.println("putObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        }
    }
}
```

```
    } catch (Exception e) {
        System.out.println("putObject failed");
        // 其他异常信息打印
        e.printStackTrace();
    }
}
```

## 代码示例：设置对象自定义元数据

本示例用于设置对象自定义元数据，用户自定义了一个名称为“property1”，值为“property-value1”的元数据和一个名称为“property2”，值为“property-value2”的元数据。将localfile上传到examplebucket桶中的objectname对象。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObjectMetadata;
import java.io.File;
public class PutObject011 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 设置对象自定义元数据
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.addUserMetadata("property1", "property-value1");
            metadata.getMetadata().put("property2", "property-value2");
            obsClient.putObject("examplebucket", "objectname", new File("localfile"), metadata);
            System.out.println("putObject successfully");
        } catch (ObsException e) {
            System.out.println("putObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("putObject failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 代码示例：设置对象存储类别

本示例用于设置对象存储类别。将本地文件localfile上传到examplebucket桶中，设置对象存储类别为低频访问存储，对象名称设置为objectname。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObjectMetadata;
import com.obs.services.model.StorageClassEnum;
import java.io.File;
public class PutObject010 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint, 此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 设置对象存储类别
            ObjectMetadata metadata = new ObjectMetadata();
            // 设置对象存储类别为低频访问存储
            metadata.setObjectStorageClass(StorageClassEnum.WARM);
            obsClient.putObject("examplebucket", "objectname", new File("localfile"), metadata);
            System.out.println("putObject successfully");
        } catch (ObsException e) {
            System.out.println("putObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("putObject failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 更多关于设置对象元数据的示例代码，请参见[Github示例](#)。
- 设置对象元数据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](#)。

## 9.7 分段上传(Java SDK)

### 9.7.1 分段上传概述(Java SDK)

#### 功能说明

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。

使用分段上传具有以下优势：

- 提高吞吐量——您可以使用并行上传段以提高吞吐量。
- 从任何网络问题中快速恢复——较小的段大小可以将由于网络错误而需重启失败的上传所产生的影响降至最低。
- 暂停和恢复对象上传——您可以随时上传对象段。启动多段上传后，不存在过长期限；您必须显式地完成或取消多段上传任务。
- 在得知最终对象大小之前开始上传——您可以在创建对象的同时上传对象。

分段上传分为如下3个步骤：

- 步骤1 **初始化分段上传任务**（`ObsClient.initiateMultipartUpload`）。
- 步骤2 **逐个或并行上传段**（`ObsClient.uploadPart`）。
- 步骤3 **合并段**（`ObsClient.completeMultipartUpload`）或**取消分段上传任务**（`ObsClient.abortMultipartUpload`）。

----结束

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 9.7.2 初始化分段上传任务(Java SDK)

#### 功能说明

使用分段上传方式传输数据前，必须先通知OBS初始化一个分段上传任务。该操作会返回一个OBS服务端创建的全局唯一标识（Upload ID），用于标识本次分段上传任务。您可以根据这个唯一标识来发起相关的操作，如取消分段上传任务、列举分段上传任务、列举已上传的段等。

## 接口约束

- 您必须是桶拥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以及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，否则碎片会占用您的存储空间并产生一定的存储费用。

## 方法定义

obsClient.initiateMultipartUpload([InitiateMultipartUploadRequest request](#))

## 请求参数说明

表 9-93 initiateMultipartUpload 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">InitiateMultipartUploadRequest</a>	必选	<b>参数解释：</b> 初始化分段上传任务，详见 <a href="#">InitiateMultipartUploadRequest</a> 。

表 9-94 InitiateMultipartUploadRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
metadata	<b>ObjectMetadata</b>	可选	<p><b>参数解释:</b> 对象元数据，详见<b>ObjectMetadata</b>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 创桶时可指定桶的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>
sseKmsHeader	<a href="#">SseKmsHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseKmsHeader</a>。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseCHeader</a>。</p> <p><b>默认取值:</b> 无</p>
extensionPermissionMap	Map< <a href="#">ExtensionObjectPermissionEnum</a> , Set<String>>	可选	<p><b>参数解释:</b> 桶ACL的授权Map，您可以为一个或多个账号授予桶权限。Map的<a href="#">ExtensionObjectPermissionEnum</a>用于指定权限，Map的Set&lt;String&gt;用于说明该权限授予的账号ID列表，即domain_id列表。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 授予权限的范围详见<a href="#">ExtensionObjectPermissionEnum</a>。</li> <li>• 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</li> </ul> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
expires	int	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 此字段对于每个对象支持上传时配置，也支持后期通过修改元数据接口附带头域x-obs-expires修改。</p> <p><b>取值范围:</b> 大于0的整型数，单位：天。</p> <p><b>默认取值:</b> 无</p>
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否开启OBS对请求头域的自动编码。</p> <p>由于HTTP编码规范限制，无法发送非ASCII码字符，SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码，发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”，则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围:</b> true：启用SDK编码。 false：不启用SDK编码。</p> <p><b>默认取值:</b> true</p>

表 9-95 ACL 预定义访问策略

常量名	说明
AccessControllist.REST_CANNED_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	说明
AccessControlList.REST_CANNED_PUBLIC_READ	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。 设在对象上，所有人可以获取该对象内容和元数据。
AccessControlList.REST_CANNED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControlList.REST_CANNED_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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 9-96 ExtensionObjectPermissionEnum

常量值	说明
GRANT_READ	授权指定租户有读对象和获取对象元数据的权限。
GRANT_READ_ACP	授权指定租户有获取对象ACL的权限。
GRANT_WRITE_ACP	授权指定租户有写对象ACL的权限。
GRANT_FULL_CONTROL	授权指定租户有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。

表 9-97 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式, 对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值: AES256, 即选择SSE-C方式, 使用高级加密标准 ( Advanced Encryption Standard, AES ) 加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过Base64编码后得到, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 9-98 SseKmsHeader

参数名称	参数类型	是否必选	描述
encryption	<a href="#">ServerEncryption</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p><b>取值范围:</b> 可选值: kms, 即选择SSE-KMS方式加密对象, 详见<a href="#">ServerEncryption</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持kms。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
kmsKeyId	String	可选	<p><b>参数解释:</b> SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p><b>取值范围:</b> 有效值支持两种格式:</p> <ol style="list-style-type: none"><li>regionID:domainID(账号ID):key/key_id</li><li>key_id</li></ol> <p>其中:</p> <ul style="list-style-type: none"><li>regionID是使用密钥所属region的ID, 可在<a href="#">地区和终端节点</a>页面获取;</li><li>domainID是使用密钥所属账号的账号ID, 获取方法参见<a href="#">如何获取账号ID和用户ID?</a>;</li><li>key_id是从数据加密服务创建的密钥ID, 获取方法请参见<a href="#">查看密钥</a>。</li></ul> <p><b>默认取值:</b></p> <ul style="list-style-type: none"><li>如果用户没有提供该头域, 那么默认的主密钥将会被使用。</li><li>如果默认主密钥不存在, 将默认创建并使用。</li></ul>

表 9-99 ServerAlgorithm

常量值	原始值
AES256	AES256

表 9-100 ServerEncryption

常量值	原始值
OBS_KMS	kms

表 9-101 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 9-102 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 9-103 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 单次上传对象大小范围是[0, 5GB]。</li> <li>• 如果需要上传超过5GB的大文件，需要通过<a href="#">多段操作</a>来分段上传。</li> </ul> <p><b>默认取值:</b> 如果不设置，则SDK会自动计算对象数据的长度。</p>

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentTypeLanguage	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
expires	String	可选	<p><b>参数解释:</b> 指对象在网页中的缓存过期时间。</p> <p><b>约束限制:</b> 日期格式为GMT的格式。</p> <p><b>默认取值:</b> 无</p>
contentMd5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>对象数据的MD5值必须经过Base64编码。</li> <li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li> </ul> <p><b>取值范围:</b> 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：n58lG6hfM7vql4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>
storageClasses	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li><li>• 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。</li></ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 9-104 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	Owner	可选	<p><b>参数解释:</b> 桶所有者的信息，详见<a href="#">Owner</a>。</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>

参数名称	参数类型	是否必选	参数类型
grants	Set< <a href="#">GrantAndPermission</a> >	可选	<b>参数解释:</b> 被授权用户相关信息, 详见 <a href="#">GrantAndPermission</a> 。

表 9-105 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-106 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组, 详见 <a href="#">GranteeInterface</a> 。

参数名称	参数类型	是否必选	描述
permission	Permission	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见Permission。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 9-107 Permission

常量名	原始值	说明
PERMISSION_READ	READ	<p>读权限。</p> <p>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。</p> <p>如果有对象的读权限，则可以获取该对象内容和元数据。</p>
PERMISSION_WRITE	WRITE	<p>写权限。</p> <p>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。</p> <p>此权限在对象上不适用。</p>
PERMISSION_READ_ACP	READ_ACP	<p>读取ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有读对应桶或对象ACP的权限。</p>

常量名	原始值	说明
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

表 9-108 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<p><b>参数解释：</b> 被授权用户的用户信息，详见<a href="#">CanonicalGrantee</a>。</p>
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<p><b>参数解释：</b> 被授权用户组的用户组信息。</p> <p><b>取值范围：</b> 详见<a href="#">GroupGrantee</a>。</p> <p><b>默认取值：</b> 无</p>

表 9-109 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-110 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

## 返回结果

表 9-111 返回值类型 InitiateMultipartUploadResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
uploadId	String	<b>参数解释:</b> OBS服务端创建的全局唯一标识（Upload ID），用于标识本次分段上传任务 <b>取值范围:</b> 长度大于0且不超过32的字符串。 <b>默认取值:</b> 无
bucketName	String	<b>参数解释:</b> 分段上传任务的桶名。 <b>约束限制:</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <b>默认取值:</b> 无

参数名称	参数类型	描述
objectKey	String	<p><b>参数解释:</b> 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。</p> <p>例如，您对象的访问地址为 examplebucket.obs.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于对examplebucket桶中的objectname对象做多段上传初始化，并获取uploadId。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.InitiateMultipartUploadRequest;
import com.obs.services.model.InitiateMultipartUploadResult;
import com.obs.services.model.ObjectMetadata;
public class InitiateMultipartUpload001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            InitiateMultipartUploadRequest request = new InitiateMultipartUploadRequest("examplebucket",
"objectname");
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.addUserMetadata("property", "property-value");
            metadata.setContentType("text/plain");
            request.setMetadata(metadata);
            InitiateMultipartUploadResult result = obsClient.initiateMultipartUpload(request);
            String uploadId = result.getUploadId();
            System.out.println("initiateMultipartUpload successfully");
            System.out.println("uploadId:" + uploadId);
        } catch (ObsException e) {
```

```
System.out.println("initiateMultipartUpload failed");  
// 请求失败,打印http状态码  
System.out.println("HTTP Code:" + e.getResponseCode());  
// 请求失败,打印服务端错误码  
System.out.println("Error Code:" + e.getErrorCode());  
// 请求失败,打印详细错误信息  
System.out.println("Error Message:" + e.getErrorMessage());  
// 请求失败,打印请求id  
System.out.println("Request ID:" + e.getErrorRequestId());  
System.out.println("Host ID:" + e.getErrorHostId());  
e.printStackTrace();  
} catch (Exception e) {  
    System.out.println("initiateMultipartUpload failed");  
    // 其他异常信息打印  
    e.printStackTrace();  
}  
}
```

## 相关链接

- 关于分段上传-初始化分段上传任务的API说明, 请参见[初始化上传段任务](#)。
- 更多关于分段上传的示例代码, 请参见[Github示例](#)。
- 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](#)。

## 9.7.3 上传段(Java 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。除了最后一段以外, 其他段的大小范围是100KB~5GB; 最后一段的大小范围是0~5GB。上传的段的编号也有范围限制, 其范围是1~10000。

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上传段的权限, 才能上传段。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以及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, 否则碎片会占用您的存储空间并产生一定的存储费用。
- 段任务中的partNumber是唯一的, 重复上传相同partNumber的段, 后一次上传会覆盖前一次上传内容。多并发上传同一对象的同一partNumber时, 服务端遵循Last Write Win策略, 但“Last Write”的时间定义为段元数据创建时间。为了保证数据准确性, 客户端需要加锁保证同一对象的同一个段上传的并发性。同一对象的不同段并发上传不需要加锁。



## 方法定义

obsClient.uploadPart([UploadPartRequest request](#))

## 请求参数说明

表 9-112 uploadPart 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">UploadPartRequest</a>	必选	<b>参数解释:</b> 上传段请求参数, 详见 <a href="#">UploadPartRequest</a> 。

表 9-113 UploadPartRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释:</b> 桶名。 <b>约束限制:</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下:<ul style="list-style-type: none"><li>3~63个字符, 数字或字母开头, 支持小写字母、数字、“_”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻(如: “my..bucket”)。</li><li>禁止“.”和“-”相邻(如: “my-.bucket”和“my-bucket”)。</li></ul></li><li>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。</li></ul> <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无
partNumber	int	必选	<b>参数解释:</b> 段号。 <b>取值范围:</b> [1,10000]，超出这个范围，OBS将返回400 Bad Request错误。 <b>默认取值:</b> 无
uploadId	String	必选	<b>参数解释:</b> 分段上传任务的ID。任务ID可以通过 <a href="#">初始化分段上传任务</a> 生成。例如： 000001648453845DBB78F2340DD460D8。 <b>约束限制:</b> 长度为32的字符串。 <b>默认取值:</b> 无
input	java.io.InputStream	可选	<b>参数解释:</b> 待上传对象的数据流。 <b>约束限制:</b> file为null，需要设置input字段不为空；input为null，需要设置file字段不为空。二者需要选其一。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
file	java.io.File	可选	<b>参数解释:</b> 待上传对象的文件。 <b>约束限制:</b> file为null, 需要设置input字段不为空; input为null, 需要设置file字段不为空。二者必须选其一。 <b>默认取值:</b> 无
offset	long	可选	<b>参数解释:</b> 源文件中某一分段的起始偏移大小。 <b>取值范围:</b> 非负整数, 不大于待上传对象的大小, 单位: 字节。 <b>默认取值:</b> 0
partSize	Long	可选	<b>参数解释:</b> 当前段的长度。 <b>约束限制:</b> <ul style="list-style-type: none"><li>上传段接口要求除最后一段以外, 其他的段大小都要大于100KB。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段), 只有调用合并段接口时才会校验。</li><li>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。</li></ul> <b>取值范围:</b> 100KB~5GB, 单位: 字节。 <b>默认取值:</b> 102400字节
sseCHeader	<b>SseCHeader</b>	可选	<b>参数解释:</b> 服务端加密头域。详见 <b>SseCHeader</b> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
attachMd5	boolean	可选	<p><b>参数解释：</b> 是否自动计算待上传数据的MD5值。为了保证数据在网络传输过程中不出现错误，可以通过设置UploadPartRequest.setAttachMd5为true来让SDK自动计算每段数据的MD5值（仅在数据源为本地文件时有效），并放到Content-MD5请求头中。OBS服务端会计算上传数据的MD5值与SDK计算的MD5值比较，保证数据完整性。</p> <p><b>约束限制：</b> attachMd5和contentMd5同时使用时，忽略attachMd5字段。</p> <p><b>取值范围：</b> true：自动计算上传数据的MD5值。 false：不自动计算上传数据的MD5值。</p> <p><b>默认取值：</b> false</p>
contentMd5	String	可选	<p><b>参数解释：</b> 待上传段数据的MD5值（经过Base64编码），是上传段数据内容的唯一标识，可以通过该值识别对象内容是否有变化。</p> <p><b>约束限制：</b> attachMd5和contentMd5同时使用时，忽略attachMd5字段。</p> <p><b>取值范围：</b> 长度为32的字符串。</p> <p><b>默认取值：</b> 无</p>
progressListener	<b>ProgressListener</b>	可选	<p><b>参数解释：</b> 上传进度，详见<b>ProgressListener</b>。</p>
progressInterval	long	可选	<p><b>参数解释</b> 上传进度反馈间隔。例子：1024 * 1024L，每上传1MB数据反馈上传进度。</p> <p><b>默认取值：</b> 100 * 1024L，单位：字节。</p>

参数名称	参数类型	是否必选	描述
autoClose	boolean	可选	<p><b>参数解释:</b> 上传完成后, 自动关闭数据流。</p> <p><b>取值范围:</b> true: 自动关闭数据流。 false: 不自动关闭数据流。</p> <p><b>默认取值:</b> true</p>

表 9-114 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式, 对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值: AES256, 即选择SSE-C方式, 使用高级加密标准 (Advanced Encryption Standard, AES) 加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseCKeyBase64	String	可选	<b>参数解释:</b> SSE-C方式下的密钥，由原始密钥经过Base64编码后得到，该密钥用于加密对象。 <b>默认取值:</b> 无

表 9-115 ServerAlgorithm

常量值	原始值
AES256	AES256

表 9-116 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 9-117 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 9-118 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释:</b> 获取上传进度，详见 <a href="#">progressChanged</a> 。 <b>默认取值:</b> 无

表 9-119 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释:</b> 传输进度数据, 详见 <a href="#">ProgressStatus</a> 。 <b>默认取值:</b> 无

表 9-120 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

## 返回结果说明

表 9-121 UploadPartResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
partNumber	int	<p><b>参数解释:</b> 段号。</p> <p><b>取值范围:</b> [1,10000]</p> <p><b>默认取值:</b> 无</p>
etag	String	<p><b>参数解释:</b> 段的Base64编码的128位MD5摘要。ETag是段内容的唯一标识，可以通过该值识别段内容是否有变化。OBS会将服务端收到段数据的ETag值（段数据的MD5值）返回给用户。比如上传段时ETag为A，下载段时ETag为B，则说明段内容发生了变化。ETag只反映变化的内容，而不是其元数据。上传的段或拷贝操作创建的段，都有唯一的ETag。</p> <p><b>约束限制:</b> 当对象是服务端加密的对象时，ETag值不是对象的MD5值。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于通过obsClient.uploadPart上传段到examplebucket桶中的objectname对象。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.PartEtag;
import com.obs.services.model.UploadPartRequest;
import com.obs.services.model.UploadPartResult;
import java.io.File;
import java.util.ArrayList;
import java.util.List;
public class UploadPart001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
```



```
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
// 【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/SecurityToken，也可以使用其他外部引入方式传入。
String securityToken = System.getenv("SECURITY_TOKEN");
// endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    String uploadId = "upload id from initiateMultipartUpload";
    List<PartEtag> partEtags = new ArrayList<PartEtag>();
    // 上传第一段
    UploadPartRequest request = new UploadPartRequest("examplebucket", "objectname");
    // 设置Upload ID
    request.setUploadId(uploadId);
    // 设置分段号，范围是1~10000，
    request.setPartNumber(1);
    // 设置将要上传的大文件
    request.setFile(new File("localfile"));
    // 设置分段大小
    request.setPartSize(5 * 1024 * 1024L);
    UploadPartResult result = obsClient.uploadPart(request);
    partEtags.add(new PartEtag(result.getEtag(), result.getPartNumber()));
    // 上传第二段
    request = new UploadPartRequest("examplebucket", "objectname");
    // 设置Upload ID
    request.setUploadId(uploadId);
    // 设置分段号
    request.setPartNumber(2);
    // 设置将要上传的大文件
    request.setFile(new File("localfile"));
    // 设置第二段的段偏移量
    request.setOffset(5 * 1024 * 1024L);
    // 设置分段大小
    request.setPartSize(5 * 1024 * 1024L);
    result = obsClient.uploadPart(request);
    partEtags.add(new PartEtag(result.getEtag(), result.getPartNumber()));
    System.out.println("uploadPart successfully");
} catch (ObsException e) {
    System.out.println("uploadPart failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("uploadPart failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于分段上传-上传段的API说明，请参见[上传段](#)。
- 更多关于分段上传的示例代码，请参见[Github示例](#)。
- 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](#)。

## 9.7.4 合并段(Java SDK)

### 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拷贝到对象元数据中，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”操作制定的段列表中，“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、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以及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。

### 方法定义

obsClient.completeMultipartUpload([CompleteMultipartUploadRequest request](#))

## 请求参数说明

表 9-122 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">CompleteMultipartUploadRequest</a>	必选	<b>参数解释:</b> 合并段请求参数, 详见 <a href="#">CompleteMultipartUploadRequest</a> 。

表 9-123 CompleteMultipartUploadRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下: <ul style="list-style-type: none"> <li>3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻(如: “my..bucket” )。</li> <li>禁止“.”和“-”相邻(如: “my-.bucket”和“my.-bucket” )。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
partEtag	List<PartEtag>	必选	<p><b>参数解释:</b> 待合并的段列表。详情参见 <a href="#">PartEtag</a>。</p>
uploadId	String	必选	<p><b>参数解释:</b> 分段上传任务的ID，例如：000001648453845DBB78F2340DD460D8</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
encodingType	String	可选	<p><b>参数解释:</b> 对响应中的对象名objectKey进行指定类型的编码。如果objectKey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objectKey进行编码。</p> <p><b>取值范围:</b> 可选值为url。</p> <p><b>默认取值:</b> 无，不设置则不编码。</p>

参数名称	参数类型	是否必选	描述
userHeaders	HashMap <String, String>	可选	<p><b>参数解释:</b> 用户头域列表。HashMap中第一个String代表用户头域名称，第二个String代表用户头域对应的取值。通过每个SDK设置的用户头域userHeader透传给服务端，使SDK不做处理，让后续使用更灵活。</p> <p><b>默认取值:</b> 无</p>

表 9-124 PartEtag

参数名称	参数类型	是否必选	描述
etag	String	是	<p><b>参数解释:</b> 段的ETag值。分段的Base64编码的128位MD5摘要。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
partNumber	Integer	是	<p><b>参数解释:</b> 段号。分段号可以是不连续的。</p> <p><b>取值范围:</b> 取值范围是[1,10000]的非负整数。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 9-125 CompleteMultipartUploadResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
etag	String	<p><b>参数解释:</b> 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。上传的对象或拷贝操作创建的对象，都有唯一的ETag。</p> <p><b>约束限制:</b> 当对象是服务端加密的对象时，ETag值不是对象的MD5值，详情可参考<a href="#">合并段</a>。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 合并段所在的桶的桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 合并段后得到的对象名。 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
location	String	<p><b>参数解释:</b> 合并段后得到的对象的url。 例如：https://example-Bucket.obs.regions.myhuaweicloud.com/example-Object</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
versionId	String	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
objectUrl	String	<p><b>参数解释:</b> 合并段后得到的对象的全路径。</p> <p><b>默认取值:</b> 无</p>
encodingType	String	<p><b>参数解释:</b> 用于指定对响应中的对象名objectKey进行指定类型的编码。如果objectKey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objectKey进行编码。</p> <p><b>取值范围:</b> 可选值为url。</p> <p><b>默认取值:</b> 无，不设置则不编码。</p>

## 代码示例

本示例用于通过接口ObsClient.completeMultipartUpload 利用字段uploadId、partEtags 合并段于桶examplebucket下的objectname对象中：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.CompleteMultipartUploadRequest;
import com.obs.services.model.PartEtag;
import java.util.ArrayList;
import java.util.List;
public class CompleteMultipartUpload001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
```



```
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    String uploadId = "upload id from initiateMultipartUpload";
    List<PartEtag> partEtags = new ArrayList<PartEtag>();
    // 第一段
    PartEtag part1 = new PartEtag();
    part1.setPartNumber(1);
    part1.setEtag("etag1");
    partEtags.add(part1);
    // 第二段
    PartEtag part2 = new PartEtag();
    part2.setPartNumber(2);
    part2.setEtag("etag2");
    partEtags.add(part2);
    CompleteMultipartUploadRequest request =
        new CompleteMultipartUploadRequest("examplebucket", "objectname", uploadId, partEtags);
    obsClient.completeMultipartUpload(request);
    System.out.println("completeMultipartUpload successfully");
} catch (ObsException e) {
    System.out.println("CompleteMultipartUpload failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("completeMultipartUpload failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于分段上传-合并段的API说明, 请参见[合并段](#)。
- 更多关于分段上传的示例代码, 请参见[Github示例](#)。
- 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](#)。

### 9.7.5 分段上传场景化示例(Java SDK)

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。通过使用 `UploadPartRequest.setOffset` 和 `UploadPartRequest.setPartSize` 来设置每段的起始结束位置。下面的示例代码展示了如何使用分段上传并发上传大文件。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.CompleteMultipartUploadRequest;
import com.obs.services.model.InitiateMultipartUploadRequest;
import com.obs.services.model.InitiateMultipartUploadResult;
import com.obs.services.model.PartEtag;
import com.obs.services.model.UploadPartRequest;
import com.obs.services.model.UploadPartResult;
import java.io.File;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.concurrent.ExecutorService;
```

```
import java.util.concurrent.Executors;
import java.util.concurrent.TimeUnit;
public class ConcurrentUploadPart001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            final String bucketName = "examplebucket";
            final String objectKey = "objectname";
            // 初始化线程池
            ExecutorService executorService = Executors.newFixedThreadPool(20);
            final File largeFile = new File("localfile");
            // 初始化分段上传任务
            InitiateMultipartUploadRequest request = new InitiateMultipartUploadRequest(bucketName,
objectKey);
            InitiateMultipartUploadResult result = obsClient.initiateMultipartUpload(request);
            final String uploadId = result.getUploadId();
            System.out.println("\t" + uploadId + "\n");
            // 每段上传100MB
            long partSize = 100 * 1024 * 1024L;
            long fileSize = largeFile.length();
            // 计算需要上传的段数
            long partCount = fileSize % partSize == 0 ? fileSize / partSize : fileSize / partSize + 1;
            final List<PartEtag> partEtags = Collections.synchronizedList(new ArrayList<PartEtag>());
            // 执行并发上传段
            for (int i = 0; i < partCount; i++) {
                // 分段在文件中的起始位置
                final long offset = i * partSize;
                // 分段大小
                final long currPartSize = (i + 1 == partCount) ? fileSize - offset : partSize;
                // 分段号
                final int partNumber = i + 1;
                executorService.execute(
                    new Runnable() {
                        @Override
                        public void run() {
                            UploadPartRequest uploadPartRequest = new UploadPartRequest();
                            uploadPartRequest.setBucketName(bucketName);
                            uploadPartRequest.setObjectKey(objectKey);
                            uploadPartRequest.setUploadId(uploadId);
                            uploadPartRequest.setFile(largeFile);
                            uploadPartRequest.setPartSize(currPartSize);
                            uploadPartRequest.setOffset(offset);
                            uploadPartRequest.setPartNumber(partNumber);
                            UploadPartResult uploadPartResult;
                            try {
                                uploadPartResult = obsClient.uploadPart(uploadPartRequest);
                                System.out.println("Part#" + partNumber + " done\n");
                                partEtags.add(
                                    new PartEtag(uploadPartResult.getEtag(),
uploadPartResult.getPartNumber()));

```

```
        } catch (ObsException e) {
            e.printStackTrace();
        }
    }
});
}
// 等待上传完成
executorService.shutdown();
while (!executorService.isTerminated()) {
    try {
        executorService.awaitTermination(5, TimeUnit.SECONDS);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
// 合并段
CompleteMultipartUploadRequest completeMultipartUploadRequest =
    new CompleteMultipartUploadRequest(bucketName, objectKey, uploadId, partEtags);
obsClient.completeMultipartUpload(completeMultipartUploadRequest);
System.out.println("completeMultipartUpload successfully");
} catch (ObsException e) {
    System.out.println("CompleteMultipartUpload failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("completeMultipartUpload failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 9.7.6 取消分段上传任务(Java SDK)

### 功能说明

通过分段上传任务的ID，取消指定桶中的分段上传任务。

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以及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([AbortMultipartUploadRequest request](#))

## 请求参数

表 9-126 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">AbortMultipartUploadRequest</a>	必选	<b>参数解释:</b> 取消分段请求参数, 详见 <a href="#">AbortMultipartUploadRequest</a> 。

表 9-127 AbortMultipartUploadRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	桶名。 <b>参数解释:</b> 桶名。 <b>约束限制:</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下:<ul style="list-style-type: none"><li>3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻(如: “my..bucket”)。</li><li>禁止“.”和“-”相邻(如: “my-.bucket”和“my.-bucket”)。</li></ul></li><li>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。</li></ul> <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
uploadId	String	必选	<p><b>参数解释:</b> 分段上传任务的ID，例如： 000001648453845DBB78F2340DD460D8</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 9-128 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

通过ObsClient.abortMultipartUpload利用初始化多段上传时获取的uploadId，取消对examplebucket桶中的objectname对象分段上传任务。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AbortMultipartUploadRequest;
public class AbortMultipartUpload001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            String uploadId = "upload id from initiateMultipartUpload";
            // 取消分段上传任务请求
            AbortMultipartUploadRequest request = new AbortMultipartUploadRequest("examplebucket",
"objectname", uploadId);
            // 取消分段上传任务接口
            obsClient.abortMultipartUpload(request);
            System.out.println("AbortMultipartUpload successfully");
        } catch (ObsException e) {
            System.out.println("AbortMultipartUpload failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("AbortMultipartUpload failed");
            // 其他异常信息打印
        }
    }
}
```

```
e.printStackTrace();  
    }  
}  
}
```

## 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](#)。
- 更多关于分段上传的示例代码，请参见[Github示例](#)。
- 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](#)。

## 9.7.7 列举已上传的段(Java SDK)

### 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特定多段上传任务中正在进行上传的段信息，单次最多列举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以及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.listParts([ListPartsRequest request](#))

### 请求参数说明

表 9-129 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">ListPartsRequest</a>	必选	<b>参数解释：</b> 列举上传段请求参数，详见 <a href="#">ListPartsRequest</a> 。

表 9-130 ListPartsRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
uploadId	String	必选	<b>参数解释:</b> 分段上传任务的ID, 例如: 000001648453845DBB78F2340DD460D8 <b>取值范围:</b> 长度大于0且不超过32的字符串。 <b>默认取值:</b> 无
maxParts	Integer	可选	<b>参数解释:</b> 列举已上传段的返回结果最大段数目, 即分页时每页中段数目。 <b>约束限制:</b> 如果该参数超出1000时, 则按照默认的1000进行处理。 <b>取值范围:</b> [1, 1000] <b>默认取值:</b> 1000
partNumberMarker	Integer	可选	<b>参数解释:</b> 列举已上传段的起始位置。 <b>约束限制:</b> 只有PartNumber数目大于该参数的段会被列出 <b>默认取值:</b> 无
encodingType	String	可选	<b>参数解释:</b> 对响应中的对象名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。 <b>取值范围:</b> 可选值为url。 <b>默认取值:</b> 无, 不设置则不编码。

## 返回结果说明

表 9-131 ListPartsResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
bucket	String	<b>参数解释:</b> 桶名。 <b>约束限制:</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <b>默认取值:</b> 无

参数名称	参数类型	描述
key	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
uploadId	String	<p><b>参数解释:</b> 分段上传任务的ID，例如： 000001648453845DBB78F2340DD460D8</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
initiator	<b>Owner</b>	<p><b>参数解释:</b> 分段上传任务的创建者，详见<b>Owner</b>。</p> <p><b>默认取值:</b> 无</p>
owner	<b>Owner</b>	<p><b>参数解释:</b> 和initiator相同，代表分段上传任务的创建者，详见<b>Owner</b>。</p> <p><b>默认取值:</b> 无</p>
storageClass	<b>StorageClassEnum</b>	<p><b>参数解释:</b> 分段上传的对象的存储类别。</p> <p><b>取值范围:</b> 存储类别取值范围参见<b>StorageClassEnum</b>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
multipartList	List<Multipart>	<b>参数解释:</b> 已上传段信息列表, 详见Multipart。 <b>默认取值:</b> 无
maxParts	Integer	<b>参数解释:</b> 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。 <b>约束限制:</b> 如果该参数超出取值范围时, 按照默认的1000进行处理。 <b>取值范围:</b> [1, 1000] <b>默认取值:</b> 1000
isTruncated	boolean	<b>参数解释:</b> 表明本次请求是否返回了全部结果。 <b>取值范围:</b> <ul style="list-style-type: none"><li>• true: 表示没有返回全部结果。</li><li>• false: 表示已返回了全部结果。</li></ul> <b>默认取值:</b> 无
partNumberMarker	String	<b>参数解释:</b> 列举段的起始位置, 与请求中的该参数对应。 <b>默认取值:</b> 无
nextPartNumberMarker	String	<b>参数解释:</b> 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。 <b>默认取值:</b> 无

表 9-132 Multipart

参数名称	参数类型	描述
partNumber	Integer	<b>参数解释:</b> 段号。 <b>取值范围:</b> 取值范围是[1,10000]的非负整数。 <b>默认取值:</b> 无
lastModified	Date	<b>参数解释:</b> 段的最后上传时间。 <b>默认取值:</b> 无
etag	String	<b>参数解释:</b> 段的ETag值。分段的base64编码的128位MD5摘要。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
size	Long	<b>参数解释:</b> 段的大小。 <b>默认取值:</b> 无

表 9-133 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 9-134 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

### 代码示例：简单列举已上传的段（最多 1000 个）

通过initiateMultipartUpload获取的uploadId对examplebucket桶下的objectname对象进行上传段的简单列举，至多列举1000个段。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListPartsRequest;
import com.obs.services.model.ListPartsResult;
import com.obs.services.model.Multipart;
public class ListParts001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            String uploadId = "upload id from initiateMultipartUpload";
            // 列举已上传的段，其中uploadId来自于initiateMultipartUpload
            ListPartsRequest request = new ListPartsRequest("examplebucket", "objectname");
            request.setUploadId(uploadId);
```

```
ListPartsResult result = obsClient.listParts(request);
for (Multipart part : result.getMultipartList()) {
    // 分段号, 上传时候指定
    System.out.println("PartNumber:" + part.getPartNumber());
    // 段数据大小
    System.out.println("Size:" + part.getSize());
    // 分段的ETag值
    System.out.println("Etag:" + part.getEtag());
    // 段的最后上传时间
    System.out.println("LastModified:" + part.getLastModified());
}
System.out.println("listParts successfully");
} catch (ObsException e) {
    System.out.println("listParts failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("listParts failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：列举所有已上传的段

由于ObsClient.listParts只能列举至多1000个段，如果列举段数量大于1000个，可采用分页全部列举。示例如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListPartsRequest;
import com.obs.services.model.ListPartsResult;
import com.obs.services.model.Multipart;
public class ListParts002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            String uploadId = "upload id from initiateMultipartUpload";
            // 列举所有已上传的段，其中uploadId来自于initiateMultipartUpload
            ListPartsRequest request = new ListPartsRequest("examplebucket", "objectname");
            request.setUploadId(uploadId);
```

```
ListPartsResult result;
do {
    result = obsClient.listParts(request);
    for (Multipart part : result.getMultipartList()) {
        // 分段号, 上传时候指定
        System.out.println("PartNumber:" + part.getPartNumber());
        // 段数据大小
        System.out.println("Size:" + part.getSize());
        // 分段的ETag值
        System.out.println("Etag:" + part.getEtag());
        // 段的最后上传时间
        System.out.println("LastModified:" + part.getLastModified());
    }
    request.setPartNumberMarker(Integer.parseInt(result.getNextPartNumberMarker()));
} while (result.isTruncated());
System.out.println("listParts successfully");
} catch (ObsException e) {
    System.out.println("listParts failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("listParts failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于分段上传-列举已上传的段的API说明, 请参见[列举已上传的段](#)。
- 更多关于分段上传的示例代码, 请参见[Github示例](#)。
- 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](#)。

## 9.7.8 列举分段上传任务(Java SDK)

### 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, 如果正在进行的多段上传任务超过1000个, 您需要发送其他请求以检索剩余的多段上传任务。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 默认情况下，允许桶所有者和多段上传任务的发起者执行此操作。除了默认情况之外，桶所有者可以允许其他委托人对桶执行ListBucketMultipartUploads操作。

## 方法定义

obsClient.listMultipartUploads([ListMultipartUploadsRequest request](#))

## 请求参数说明

表 9-135 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">ListMultipartUploadsRequest</a>	必选	<b>参数解释：</b> 列举上传段请求参数，详见 <a href="#">ListMultipartUploadsRequest</a> 。

表 9-136 ListMultipartUploadsRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释：</b> 桶名。 <b>约束限制：</b> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 桶命名规则如下： <ul style="list-style-type: none"> <li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>- 禁止使用IP地址。</li> <li>- 禁止以“-”或“.”开头及结尾。</li> <li>- 禁止两个“.”相邻（如：“my..bucket”）。</li> <li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>• 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。</li> </ul> <b>默认取值：</b> 无

参数名称	参数类型	是否必选	描述
prefix	String	可选	<p><b>参数解释:</b> 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。</p> <p><b>约束限制:</b> 长度大于 0 且不超过 1024 的字符串。</p> <p><b>默认取值:</b> 无</p>
delimiter	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度大于 0 且不超过 1024 的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
maxUploads	Integer	可选	<p><b>参数解释:</b> 列举分段上传任务的最大数目。</p> <p><b>约束限制:</b> 当该参数超出1000时, 按照默认的1000进行处理。</p> <p><b>取值范围:</b> 1~1000, 单位: 个。</p> <p><b>默认取值:</b> 1000</p>
keyMarker	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 上次请求返回体的nextKeyMarker值。</p> <p><b>默认取值:</b> 无</p>
uploadIdMarker	String	可选	<p><b>参数解释:</b> 列举分段上传任务的起始位置 (uploadId标识)。</p> <p><b>约束限制:</b> 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。</p> <p><b>取值范围:</b> 对象的分段上传任务ID, 即上次请求返回体的nextUploadIdMarker值。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
encodingType	String	可选	<p><b>参数解释:</b> 响应中的部分元素进行指定类型的编码。如果 delimiter、keyMarker、prefix、nextKeyMarker 和 objectKey 包含xml 1.0标准不支持的控制字符，可通过设置 encodingType 对响应中的 delimiter、keyMarker、prefix（包括commonPrefixes 中的 prefix）、nextKeyMarker 和 objectKey 进行编码。</p> <p><b>取值范围:</b> 可选值为url。</p> <p><b>默认取值:</b> 无，不设置则不编码。</p>

## 返回结果说明

表 9-137 MultipartUploadListing

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
bucket	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
keyMarker	String	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
uploadIdMarker	String	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
nextKeyMarker	String	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无
nextUploadIdMarker	String	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度大于0且不超过32的字符串。 <b>默认取值:</b> 无
prefix	String	<b>参数解释:</b> 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。 <b>约束限制:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	描述
maxUploads	int	<b>参数解释:</b> 列举分段上传任务的最大数目，与请求中的该参数对应。 <b>约束限制:</b> 当该参数超出范围时，按照默认的1000进行处理。 <b>取值范围:</b> 1~1000，单位：个。 <b>默认取值:</b> 1000
truncated	boolean	<b>参数解释:</b> 表明本次请求是否返回了全部结果。 <b>取值范围:</b> <ul style="list-style-type: none"><li>• true：表示没有返回全部结果。</li><li>• false：表示已返回了全部结果。</li></ul> <b>默认取值:</b> 无
multipartTaskList	List<Multipart Upload>	<b>参数解释:</b> 桶内分段上传任务列表，详见 <a href="#">MultipartUpload</a> 。

参数名称	参数类型	描述
delimiter	String	<p><b>参数解释:</b> 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。</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><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
commonPrefixes	String[]	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>



表 9-138 MultipartUpload

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
uploadId	String	<p><b>参数解释:</b> 分段上传任务的ID，例如： 000001648453845DBB78F2340DD460D8</p> <p><b>取值范围:</b> 长度大于0且不超过32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
initiatedDate	java.util.Date	<b>参数解释:</b> 分段上传任务的初始化时间。 <b>约束限制:</b> 日期格式必须为ISO8601的格式。 <b>默认取值:</b> 无
storageClass	<b>StorageClass Enum</b>	<b>参数解释:</b> 分段上传的对象的存储类别。 <b>取值范围:</b> 存储类别取值范围参见 <b>StorageClassEnum</b> 。 <b>默认取值:</b> 无
initiator	<b>Owner</b>	<b>参数解释:</b> 分段上传任务的创建者，详见 <b>Owner</b> 。
owner	<b>Owner</b>	<b>参数解释:</b> 和initiator相同，代表分段上传任务的创建者，详见 <b>Owner</b> 。

表 9-139 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 9-140 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

## 代码示例：简单列举分段上传任务

对examplebucket桶进行简单列举分段上传任务，至多返回1000个任务信息。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListMultipartUploadsRequest;
import com.obs.services.model.MultipartUpload;
import com.obs.services.model.MultipartUploadListing;
public class ListMultipartUploads001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint, 此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            ListMultipartUploadsRequest request = new ListMultipartUploadsRequest("examplebucket");
            MultipartUploadListing result = obsClient.listMultipartUploads(request);
            for (MultipartUpload upload : result.getMultipartTaskList()) {
                System.out.println("UploadId:" + upload.getUploadId());
                System.out.println("ObjectKey:" + upload.getObjectKey());
                System.out.println("InitiatedDate:" + upload.getInitiatedDate());
            }
        }
    }
}
```

```
    }
    System.out.println("ListMultipartUploads successfully");
} catch (ObsException e) {
    System.out.println("ListMultipartUploads failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("ListMultipartUploads failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：分页列举所有已上传的段任务

由于ObsClient.listMultipartUploads只能列举至多1000个任务信息，如果列举段任务信息数量大于1000个，可采用分页全部列举。示例如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListMultipartUploadsRequest;
import com.obs.services.model.MultipartUpload;
import com.obs.services.model.MultipartUploadListing;
public class ListMultipartUploads002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            ListMultipartUploadsRequest request = new ListMultipartUploadsRequest("examplebucket");
            MultipartUploadListing result;
            do {
                result = obsClient.listMultipartUploads(request);
                for (MultipartUpload upload : result.getMultipartTaskList()) {
                    System.out.println("UploadId:" + upload.getUploadId());
                    System.out.println("ObjectKey:" + upload.getObjectKey());
                    System.out.println("InitiatedDate:" + upload.getInitiatedDate());
                }
                request.setKeyMarker(result.getNextKeyMarker());
                request.setUploadIdMarker(result.getNextUploadIdMarker());
            } while (result.isTruncated());
            System.out.println("ListMultipartUploads successfully");
        } catch (ObsException e) {
```

```
System.out.println("ListMultipartUploads failed");  
// 请求失败,打印http状态码  
System.out.println("HTTP Code:" + e.getResponseCode());  
// 请求失败,打印服务端错误码  
System.out.println("Error Code:" + e.getErrorCode());  
// 请求失败,打印详细错误信息  
System.out.println("Error Message:" + e.getErrorMessage());  
// 请求失败,打印请求id  
System.out.println("Request ID:" + e.getErrorRequestId());  
System.out.println("Host ID:" + e.getErrorHostId());  
e.printStackTrace();  
} catch (Exception e) {  
    System.out.println("ListMultipartUploads failed");  
    // 其他异常信息打印  
    e.printStackTrace();  
}  
}
```

## 相关链接

- 关于分段上传-列举分段上传任务的API说明，请参见[列举桶中已初始化多段任务](#)。
- 更多关于分段上传的示例代码，请参见[Github示例](#)。
- 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](#)。

## 9.8 设置对象生命周期(Java SDK)

### 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](#)。此接口设置的对象过期时间，其优先级高于桶生命周期规则。

上传对象或者初始化分段上传任务时，您可以直接指定对象的过期时间。

#### 📖 说明

- 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。如果对象经过转换，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。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 此接口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。

## 方法定义

obsClient.putObject([PutObjectRequest request](#))

## 上传对象请求参数说明

表 9-141 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">PutObjectRequest</a>	必选	<b>参数解释：</b> 上传对象请求参数，详见 <a href="#">PutObjectRequest</a> 。

表 9-142 PutObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
metadata	<b>ObjectMetadata</b>	可选	<p><b>参数解释:</b> 对象元数据，详见<a href="#">ObjectMetadata</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 创桶时可指定桶的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List，ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>
sseKmsHeader	<a href="#">SseKmsHeader</a>	可选	<p><b>参数解释:</b> SSE-KMS服务端加密头信息。详见<a href="#">SseKmsHeader</a>。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> SSE-C服务端加密头信息。详见<a href="#">SseCHeader</a>。</p> <p><b>默认取值:</b> 无</p>
input	java.io.InputStream	可选	<p><b>参数解释:</b> 待上传对象的数据流。</p> <p><b>默认取值:</b> 无</p>
file	java.io.File	可选	<p><b>参数解释:</b> 待上传对象的文件。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
extensionPermissionMap	Map<ExtensionObjectPermissionEnum, Set<String>>	可选	<p><b>参数解释:</b> 桶ACL的授权Map，您可以为一个或多个账号授予桶权限。Map的ExtensionObjectPermissionEnum用于指定权限，Map的Set&lt;String&gt;用于说明该权限授予的账号ID列表，即domain_id列表。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>授予权限的范围详见ExtensionObjectPermissionEnum。</li> <li>如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</li> </ul> <p><b>默认取值:</b> 无</p>
expires	int	可选	<p><b>参数解释:</b> 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><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>此字段对于每个对象支持上传时配置，也支持后期通过修改元数据接口附带头域x-obs-expires修改。</li> <li>对象生命周期到期以后，对象将会永久删除，无法恢复。</li> </ul> <p><b>取值范围:</b> 大于0的整型数，单位：天。</p> <p><b>默认取值:</b> 无</p>
progressListener	ProgressListener	可选	<p><b>参数解释:</b> 上传进度，详见<a href="#">ProgressListener</a>。</p>

参数名称	参数类型	是否必选	描述
encodeHeaders	boolean	可选	<p><b>参数解释：</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制，无法发送非ASCII码字符，SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码，发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”，则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围：</b> true：启用SDK编码。 false：不启用SDK编码。</p> <p><b>默认取值：</b> true</p>

表 9-143 ACL 预定义访问策略

常量名	说明
AccessControllist.REST_CANNED_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。
AccessControllist.REST_CANNED_PUBLIC_READ	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。 设在对象上，所有人可以获取该对象内容和元数据。
AccessControllist.REST_CANNED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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	说明
AccessControlList.REST_CANNED_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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 9-144 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释：</b> 获取上传进度，详见 <a href="#">progressChanged</a> 。 <b>默认取值：</b> 无

表 9-145 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释：</b> 传输进度数据，详见 <a href="#">ProgressStatus</a> 。 <b>默认取值：</b> 无

表 9-146 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。

方法名称	返回值类型	说明
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 9-147 ExtensionObjectPermissionEnum

常量值	说明
GRANT_READ	授权指定租户有读对象和获取对象元数据的权限。
GRANT_READ_ACP	授权指定租户有获取对象ACL的权限。
GRANT_WRITE_ACP	授权指定租户有写对象ACL的权限。
GRANT_FULL_CONTROL	授权指定租户有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。

表 9-148 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard，AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过Base64编码后得到, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 9-149 SseKmsHeader

参数名称	参数类型	是否必选	描述
encryption	<a href="#">ServerEncryption</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p><b>取值范围:</b> 可选值: kms, 即选择SSE-KMS方式加密对象, 详见<a href="#">ServerEncryption</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持kms。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
kmsKeyId	String	可选	<p><b>参数解释:</b> SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p><b>取值范围:</b> 有效值支持两种格式:</p> <ol style="list-style-type: none"> <li>regionID:domainID(账号ID):key/key_id</li> <li>key_id</li> </ol> <p>其中:</p> <ul style="list-style-type: none"> <li>regionID是使用密钥所属region的ID, 可在<a href="#">地区和终端节点</a>页面获取;</li> <li>domainID是使用密钥所属账号的账号ID, 获取方法参见<a href="#">如何获取账号ID和用户ID?</a>;</li> <li>key_id是从数据加密服务创建的密钥ID, 获取方法请参见<a href="#">查看密钥</a>。</li> </ul> <p><b>默认取值:</b></p> <ul style="list-style-type: none"> <li>如果用户没有提供该头域, 那么默认的主密钥将会被使用。</li> <li>如果默认主密钥不存在, 将默认创建并使用。</li> </ul>

表 9-150 ServerAlgorithm

常量值	原始值
AES256	AES256

表 9-151 ServerEncryption

常量值	原始值
OBS_KMS	kms

表 9-152 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 9-153 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 9-154 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 单次上传对象大小范围是[0, 5GB]。</li> <li>• 如果需要上传超过5GB的大文件, 需要通过<a href="#">多段操作</a>来分段上传。</li> </ul> <p><b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。</p>

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentTypeLanguage	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
expires	String	可选	<p><b>参数解释:</b> 指对象在网页中的缓存过期时间。</p> <p><b>约束限制:</b> 日期格式为GMT的格式。</p> <p><b>默认取值:</b> 无</p>
contentMd5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <p><b>取值范围:</b> 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：n58lG6hfM7vql4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>
storageClasses	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>约束限制:</b> 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。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 9-155 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	<b>Owner</b>	可选	<p><b>参数解释:</b> 桶所有者的信息，详见<a href="#">Owner</a>。</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>
grants	Set< <a href="#">GrantAndPermission</a> >	可选	<p><b>参数解释:</b> 被授权用户相关信息，详见<a href="#">GrantAndPermission</a>。</p>

表 9-156 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-157 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组, 详见 <a href="#">GranteeInterface</a> 。
permission	<a href="#">Permission</a>	必选	<b>参数解释:</b> 用户或用户组被授予的权限。 <b>取值范围:</b> 详见 <a href="#">Permission</a> 。 <b>默认取值:</b> 无
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false

表 9-158 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。
PERMISSION_WRITE_ACP	WRITE_ACP	修改ACL配置的权限。 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。
PERMISSION_FULL_CONTROL	FULL_CONTROL	完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。 如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。 如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。

表 9-159 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 9-160 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-161 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

## 返回结果说明

表 9-162 PutObjectResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
storageClass	<a href="#">StorageClass Enum</a>	<b>参数解释:</b> 对象的存储类别，当对象存储类别是标准存储时，该值为空。 <b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无
versionId	String	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	描述
etag	String	<p><b>参数解释:</b> 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。上传的对象或拷贝操作创建的对象, 都有唯一的etag。</p> <p><b>约束限制:</b> 当对象是服务端加密的对象时, etag值不是对象的MD5值。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 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.cn- north-4.myhuaweicloud.com/folder/test.txt 中, 对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例通过将localfile上传到examplebucket桶下的objectname对象中，并设置对象过期时间为30天。示例代码如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
import com.obs.services.model.PutObjectRequest;
import java.io.File;
public class PutObject012 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);
```

```
try {
    PutObjectRequest request = new PutObjectRequest();
    request.setBucketName("examplebucket");
    request.setObjectKey("objectname");
    request.setFile(new File("localfile"));
    // 上传对象时, 设置对象30天后过期
    request.setExpires(30);
    obsClient.putObject(request);
    System.out.println("putObject successfully");
} catch (ObsException e) {
    System.out.println("putObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("putObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置桶的生命周期配置的API说明, 请参见[设置桶的生命周期配置](#)。
- 设置桶的生命周期配置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](#)。
- 生命周期管理相关问题请参见[生命周期管理相关问题](#)。

## 9.9 追加上传(Java SDK)

### 功能说明

1. 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。
2. 追加对象可以实现在桶中指定的对象尾直接追加上传数据, 不存在相同名称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。
3. 每次追加上传成功后, 可通过AppendObjectResult.getNextPosition获取下次追加上传的位置; 或者通过ObsClient.getObjectMetadata接口获取下次追加上传的位置。

#### 须知

开发过程中, 您有任何问题可以在github上[提交issue](#), 或者在[华为云对象存储服务论坛](#)中发帖求助。

## 接口约束

- 您必须是桶所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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 每次追加上传的长度不能超过单次上传长度上限5G的限制。
- 每个Appendable对象追加写次数最多为10000次，如果对象追加写次数超过10000次，OBS返回409 Conflict，错误码为ObjectNotAppendable。
- 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写对象。
- 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。
- 对已有对象进行追加上传，要求对象必须为可追加上传的对象，否则将抛出异常（HTTP状态码为409）。判断对象是否可追加上传可通过[获取对象元数据](#)查询。

## 方法定义

obsClient.appendObject([AppendObjectRequest request](#))

## 请求参数说明

表 9-163 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">AppendObjectRequest</a>	必选	<b>参数解释：</b> 追加上传对象请求参数，详见 <a href="#">AppendObjectRequest</a> 。

表 9-164 AppendObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my.bucket”和“my.bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
metadata	<b>ObjectMetadata</b>	可选	<p><b>参数解释:</b> 对象元数据，详见<b>ObjectMetadata</b>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 创桶时可指定桶的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>
sseKmsHeader	<a href="#">SseKmsHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseKmsHeader</a>。</p> <p><b>约束限制:</b> 服务端加密SSE-KMS方式，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。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseCHeader</a>。</p> <p><b>约束限制:</b> 服务端加密SSE-C方式，那么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。</p> <p><b>默认取值:</b> 无</p>
input	java.io.InputStream	可选	<p><b>参数解释:</b> 待上传对象的数据流。</p> <p><b>默认取值:</b> 无</p>
file	java.io.File	可选	<p><b>参数解释:</b> 待上传对象的文件。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
extensionPermissionMap	Map<ExtensionObjectPermissionEnum, Set<String>>	可选	<p><b>参数解释:</b> 桶ACL的授权Map, 您可以为一个或多个账号授予桶权限。Map的ExtensionObjectPermissionEnum用于指定权限, Map的Set&lt;String&gt;用于说明该权限授予的账号ID列表, 即domain_id列表。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>授予权限的范围详见ExtensionObjectPermissionEnum。</li> <li>如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a></li> </ul> <p><b>默认取值:</b> 无</p>
expires	int	可选	<p><b>参数解释:</b> 表示对象的过期时间(从对象最后修改时间开始计算), 单位是天。过期之后对象会被自动删除。</p> <p><b>约束限制:</b> 此字段对于每个对象支持上传时配置, 也支持后期通过修改元数据接口附带头域x-obs-expires修改。</p> <p><b>取值范围:</b> 大于0的整型数, 单位: 天。</p> <p><b>默认取值:</b> 无</p>
progressListener	ProgressListener	可选	<p><b>参数解释:</b> 上传进度, 详见ProgressListener</p>
position	long	必选	<p><b>参数解释:</b> 追加写位置。</p> <p><b>约束限制:</b> 需要追加写的对象首次上传时必须指定position为0, 下次追加需要填写的position会在服务端本次上传成功返回消息的头域x-obs-next-append-position中携带。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制，无法发送非ASCII码字符，SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码，发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”，则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围:</b> true：启用SDK编码。 false：不启用SDK编码。</p> <p><b>默认取值:</b> true</p>

表 9-165 ACL 预定义访问策略

常量名	说明
AccessControllist.REST_CANNED_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。
AccessControllist.REST_CANNED_PUBLIC_READ	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。 设在对象上，所有人可以获取该对象内容和元数据。
AccessControllist.REST_CANNED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControllist.REST_CANNED_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，可以获取该桶内对象的内容和元数据。 不能应用于对象。

常量名	说明
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 9-166 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释：</b> 获取上传进度，详见 <a href="#">progressChanged</a> 。 <b>默认取值：</b> 无

表 9-167 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释：</b> 传输进度数据，详见 <a href="#">ProgressStatus</a> 。 <b>默认取值：</b> 无

表 9-168 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。



方法名称	返回值类型	说明
getTransferredBytes() ( )	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 9-169 ExtensionObjectPermissionEnum

常量值	说明
GRANT_READ	授权指定租户有读对象和获取对象元数据的权限。
GRANT_READ_ACP	授权指定租户有获取对象ACL的权限。
GRANT_WRITE_ACP	授权指定租户有写对象ACL的权限。
GRANT_FULL_CONTROL	授权指定租户有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。

表 9-170 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard，AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过Base64编码后得到, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 9-171 SseKmsHeader

参数名称	参数类型	是否必选	描述
encryption	<a href="#">ServerEncryption</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p><b>取值范围:</b> 可选值: kms, 即选择SSE-KMS方式加密对象, 详见<a href="#">ServerEncryption</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持kms。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
kmsKeyId	String	可选	<p><b>参数解释:</b> SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p><b>取值范围:</b> 有效值支持两种格式:</p> <ol style="list-style-type: none"> <li>regionID:domainID(账号ID):key/key_id</li> <li>key_id</li> </ol> <p>其中:</p> <ul style="list-style-type: none"> <li>regionID是使用密钥所属region的ID,可在<a href="#">地区和终端节点</a>页面获取;</li> <li>domainID是使用密钥所属账号的账号ID,获取方法参见<a href="#">如何获取账号ID和用户ID?</a>;</li> <li>key_id是从数据加密服务创建的密钥ID,获取方法请参见<a href="#">查看密钥</a>。</li> </ul> <p><b>默认取值:</b></p> <ul style="list-style-type: none"> <li>如果用户没有提供该头域,那么默认的主密钥将会被使用。</li> <li>如果默认主密钥不存在,将默认创建并使用。</li> </ul>

表 9-172 ServerAlgorithm

常量值	原始值
AES256	AES256

表 9-173 ServerEncryption

常量值	原始值
OBS_KMS	kms

表 9-174 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 9-175 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 9-176 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 单次上传对象大小范围是[0, 5GB]。</li><li>• 如果需要上传超过5GB的大文件, 需要通过<a href="#">多段操作</a>来分段上传。</li></ul> <p><b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型 ( MIME类型 )。contentType ( 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><b>取值范围:</b> 常见的contentType ( MIME ) 列表参见<a href="#">如何理解Content-Type ( MIME ) ? (Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type, SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type ( 如.xml判断为application/xml文件; .html判断为text/html文件 )。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
contentDisposition	String	可选	<b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象, 当下载对象或者访问对象时, 以默认文件名命名的文件将直接在浏览器上显示, 或在弹出文件下载对话框时显示。 <b>默认取值:</b> 无
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无
contentType	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合, 以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentMd5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <p><b>取值范围:</b> 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：n58lG6hfM7vql4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>
storageClass	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 9-177 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	<a href="#">Owner</a>	可选	<p><b>参数解释:</b> 桶所有者的信息，详见<a href="#">Owner</a>。</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>
grants	Set< <a href="#">GrantAndPermission</a> >	可选	<p><b>参数解释:</b> 被授权用户相关信息，详见<a href="#">GrantAndPermission</a>。</p>



表 9-178 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 9-179 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<p><b>参数解释:</b> 被授权用户或用户组, 详见<a href="#">GranteeInterface</a>。</p>
permission	<a href="#">Permission</a>	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 9-180 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。
PERMISSION_WRITE_ACP	WRITE_ACP	修改ACL配置的权限。 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。
PERMISSION_FULL_CONTROL	FULL_CONTROL	完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。 如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。 如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。

表 9-181 GranteeInterface

参数名称	描述
<b>CanonicalGrantee</b>	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<b>GroupGrantee</b>	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 9-182 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-183 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

## 返回结果说明

表 9-184 AppendObjectResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
etag	String	<p><b>参数解释:</b> etag值即Base64编码的128位MD5摘要，追加上传返回的etag是追加数据内容的etag，不是完整对象的etag。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。上传的对象或拷贝操作创建的对象，都有唯一的etag。</p> <p><b>约束限制:</b> 当对象是服务端加密的对象时，etag值不是对象的MD5值。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
storageClass	<b>StorageClass Enum</b>	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<b>StorageClassEnum</b>。</p> <p><b>默认取值:</b> 无</p>
nextPosition	long	<p><b>参数解释:</b> 下一次请求应该提供的position，即追加上传的位置。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于通过设置每次追加上传的不同位置将数据内容追加上传到examplebucket桶下同一个objectname对象中。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AppendObjectRequest;
import com.obs.services.model.AppendObjectResult;
import com.obs.services.model.ObjectMetadata;
import java.io.ByteArrayInputStream;
public class AppendObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try (ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint)) {
            // 第一次追加上传
            AppendObjectRequest request = new AppendObjectRequest();
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname");
            request.setPosition(0);
            request.setInput(new ByteArrayInputStream("Hello OBS".getBytes()));
            AppendObjectResult result = obsClient.appendObject(request);
        }
    }
}
```

```
// 第二次追加上传
request.setPosition(result.getNextPosition());
request.setInput(new ByteArrayInputStream("Hello OBS Again".getBytes()));
result = obsClient.appendObject(request);
System.out.println("appendObject successfully");
System.out.println("NextPosition:" + result.getNextPosition());
System.out.println("Etag:" + result.getEtag());
// 通过获取对象属性接口获取下次追加上传的位置
ObjectMetadata metadata = obsClient.getObjectMetadata("examplebucket", "objectname");
System.out.println("NextPosition from metadata:" + metadata.getNextPosition());
} catch (ObsException e) {
    System.out.println("appendObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("appendObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于上传对象-追加上传的API说明, 请参见[追加写对象](#)。
- 上传对象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](#)。
- 上传对象常见问题请参见[上传对象失败](#)。

## 9.10 断点续传上传(Java SDK)

### 功能说明

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。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, 否则返回错误信息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)。

#### 须知

开发过程中, 您有任何问题可以在github上[提交issue](#), 或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 断点续传上传接口传入的文件大小至少要100K以上。
- 使用SDK的断点续传接口时，必须开启断点续传选项后才能在进程再次进入时读取上一次上传的进度。

## 方法定义

obsClient.uploadFile([UploadFileRequest request](#))

## 请求参数说明

表 9-185 uploadFile 请求参数

参数	类型	是否必选	描述
request	<a href="#">UploadFileRequest</a>	是	<b>参数解释：</b> 上传对象请求参数，详见 <a href="#">UploadFileRequest</a> 。



表 9-186 UploadFileRequest

参数	类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数	类型	是否必选	描述
objectMetadata	<a href="#">ObjectMetadata</a>	可选	<p><b>参数解释：</b> 对象元数据，详见<a href="#">ObjectMetadata</a>。</p> <p><b>默认取值：</b> 无</p>
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释：</b> 创桶时可指定桶的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围：</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值：</b> AccessControlList.REST_CANNED_PRIVATE</p>
sseKmsHeader	<a href="#">SseKmsHeader</a>	可选	<p><b>参数解释：</b> 服务端加密头信息。详见<a href="#">SseKmsHeader</a>。</p> <p><b>默认取值：</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释：</b> 服务端加密头信息。详见<a href="#">SseCHeader</a>。</p> <p><b>默认取值：</b> 无</p>
enableCheckpoint	boolean	可选	<p><b>参数解释：</b> 是否开启断点续传模式。</p> <p><b>取值范围：</b> true：开启断点续传模式。 false：关闭断点续传模式。</p> <p><b>默认取值：</b> false</p>

参数	类型	是否必选	描述
checkpointFile	String	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 仅在断点续传模式下有效。</p> <p><b>默认取值:</b> 当该值为空时, 默认为待上传的本地文件的同级目录。</p>
uploadFile	String	必选	<p><b>参数解释:</b> 待上传文件或文件夹的完整路径, 如 aa/bb.txt, 或aa/。</p> <p><b>默认取值:</b> 无</p>
extensionPermissionMap	Map<ExtensionObjectPermissionEnum, Set<String>>	可选	<p><b>参数解释:</b> 桶ACL的授权Map, 您可以为一个或多个账号授予桶权限。Map的ExtensionObjectPermissionEnum用于指定权限, Map的Set&lt;String&gt;用于说明该权限授予的账号ID列表, 即 domain_id列表。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>授予权限的范围详见ExtensionObjectPermissionEnum。</li> <li>如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a></li> </ul> <p><b>默认取值:</b> 无</p>
progressListener	ProgressListener	可选	<p><b>参数解释:</b> 设置数据传输监听器, 用于获取上传进度。详见<a href="#">ProgressListener</a></p>
partSize	long	可选	<p><b>参数解释:</b> 分段大小。</p> <p><b>取值范围:</b> 100KB~5GB, 单位: 字节。</p> <p><b>默认取值:</b> 9MB</p>

参数	类型	是否必选	描述
taskNum	int	可选	<p><b>参数解释:</b> 分段上传时的最大并发数。</p> <p><b>取值范围:</b> 1~10000, 单位: 个。</p> <p><b>默认取值:</b> 1, 即不设置则默认单任务上传。</p>
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否开启OBS对请求头域的自动编码。</p> <p>由于HTTP编码规范限制, 无法发送非ASCII码字符, SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码, 发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”, 则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围:</b> true: 启用SDK编码。 false: 不启用SDK编码。</p> <p><b>默认取值:</b> true</p>

表 9-187 ACL 预定义访问策略

常量名	说明
AccessControllist.REST_CANNED_PRIVATE	<p>私有读写。</p> <p>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。</p>
AccessControllist.REST_CANNED_PUBLIC_READ	<p>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。</p> <p>设在对象上, 所有人可以获取该对象内容和元数据。</p>

常量名	说明
AccessControllist.REST_CAN NED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControllist.REST_CAN NED_PUBLIC_READ_DELIVER ED	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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControllist.REST_CAN NED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 9-188 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释：</b> 获取上传进度，详见 <a href="#">progressChanged</a> 。 <b>默认取值：</b> 无

表 9-189 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStat us</a>	必选	<b>参数解释：</b> 传输进度数据，详见 <a href="#">ProgressStatus</a> 。 <b>默认取值：</b> 无

表 9-190 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 9-191 ExtensionObjectPermissionEnum

常量值	说明
GRANT_READ	授权指定租户有读对象和获取对象元数据的权限。
GRANT_READ_ACP	授权指定租户有获取对象ACL的权限。
GRANT_WRITE_ACP	授权指定租户有写对象ACL的权限。
GRANT_FULL_CONTROL	授权指定租户有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。

表 9-192 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard, AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过Base64编码后得到, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 9-193 SseKmsHeader

参数名称	参数类型	是否必选	描述
encryption	<a href="#">ServerEncryption</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p><b>取值范围:</b> 可选值: kms, 即选择SSE-KMS方式加密对象, 详见<a href="#">ServerEncryption</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持kms。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
kmsKeyId	String	可选	<p><b>参数解释:</b> SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p><b>取值范围:</b> 有效值支持两种格式:</p> <ol style="list-style-type: none"> <li>regionID:domainID(账号ID):key/key_id</li> <li>key_id</li> </ol> <p>其中:</p> <ul style="list-style-type: none"> <li>regionID是使用密钥所属region的ID, 可在<a href="#">地区和终端节点</a>页面获取;</li> <li>domainID是使用密钥所属账号的账号ID, 获取方法参见<a href="#">如何获取账号ID和用户ID?</a>;</li> <li>key_id是从数据加密服务创建的密钥ID, 获取方法请参见<a href="#">查看密钥</a>。</li> </ul> <p><b>默认取值:</b></p> <ul style="list-style-type: none"> <li>如果用户没有提供该头域, 那么默认的主密钥将会被使用。</li> <li>如果默认主密钥不存在, 将默认创建并使用。</li> </ul>

表 9-194 ServerAlgorithm

常量值	原始值
AES256	AES256



表 9-195 ServerEncryption

常量值	原始值
OBS_KMS	kms

表 9-196 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 9-197 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 9-198 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<b>参数解释:</b> 对象数据的长度。 <b>约束限制:</b> <ul style="list-style-type: none"><li>• 单次上传对象大小范围是[0, 5GB]。</li><li>• 如果需要上传超过5GB的大文件，需要通过<a href="#">多段操作</a>来分段上传。</li></ul> <b>默认取值:</b> 如果不设置，则SDK会自动计算对象数据的长度。

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentTypeLanguage	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无
contentMd5	String	可选	<b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。 <b>约束限制:</b> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <b>取值范围:</b> 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：n58lG6hfM7vql4K0vnWpog== <b>默认取值:</b> 无
storageClasses	<a href="#">StorageClassEnum</a>	可选	<b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。 <b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li><li>• 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。</li></ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 9-199 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	Owner	可选	<p><b>参数解释:</b> 桶所有者的信息，详见<a href="#">Owner</a>。</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>
grants	Set<GrantAndPermission>	可选	<p><b>参数解释:</b> 被授权用户相关信息，详见<a href="#">GrantAndPermission</a>。</p>

表 9-200 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 9-201 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<p><b>参数解释:</b> 被授权用户或用户组, 详见<a href="#">GranteeInterface</a>。</p>
permission	<a href="#">Permission</a>	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 9-202 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。
PERMISSION_WRITE_ACP	WRITE_ACP	修改ACL配置的权限。 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。
PERMISSION_FULL_CONTROL	FULL_CONTROL	完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。 如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。 如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。

表 9-203 GranteeInterface

参数名称	描述
<b>CanonicalGrantee</b>	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<b>GroupGrantee</b>	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 9-204 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 9-205 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。



## 返回结果说明

表 9-206 CompleteMultipartUploadResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
etag	String	<p><b>参数解释:</b> 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。上传的对象或拷贝操作创建的对象，都有唯一的etag。</p> <p><b>约束限制:</b> 当对象是服务端加密的对象时，etag值不是对象的MD5值。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例展示了如何通过使用断点续传方式将本地文件localfile上传到examplebucket桶下的objectKey对象中。

```
import com.obs.services.ObsClient;  
import com.obs.services.exception.ObsException;  
import com.obs.services.model.CompleteMultipartUploadResult;  
import com.obs.services.model.UploadFileRequest;
```

```
public class UploadFile001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            UploadFileRequest request = new UploadFileRequest("examplebucket", "objectKey");
            // 设置待上传的本地文件，localfile为待上传的本地文件路径，需要指定到具体带文件后缀的文件名
            request.setUploadFile("localfile");
            // 设置分段上传时的最大并发数
            request.setTaskNum(5);
            // 设置分段大小为10MB
            request.setPartSize(10 * 1024 * 1024);
            // 开启断点续传模式
            request.setEnableCheckpoint(true);
            // 进行断点续传上传
            CompleteMultipartUploadResult result = obsClient.uploadFile(request);
            System.out.println("UploadFile successfully");
        } catch (ObsException e) {
            // 发生异常时可再次调用断点续传上传接口进行重新上传
            System.out.println("UploadFile failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("UploadFile failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 上传对象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](#)。
- 上传对象常见问题请参见[上传对象失败](#)。

## 9.11 基于表单上传(Java SDK)

### 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.createPostSignature生成基于表单上传的请求参数。使用代码模拟表单上传的完整代码示例，参见[PostObjectSample](#)。您也可以通过如下步骤进行表单上传：

- 步骤1** 使用ObsClient.createPostSignature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

#### 说明

使用SDK生成的用于鉴权的请求参数包括两个：

- Policy，对应表单中policy字段。
- Signature，对应表单中的signature字段。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- HTML表单中的policy，signature的值均是从ObsClient.createPostSignature的返回结果中获取。
- 上传对象 contentType字段需要手动修改为对应的Content-Type值。
- 在设置对象ACL权限时，HTML表单中的acl取值参见 [OBS预定义的权限控制策略](#)。
- 您可以直接下载表单HTML示例[PostDemo](#)。

### 方法定义

obsClient.createPostSignature([PostSignatureRequest request](#))

表 9-207 createPostSignature 请求参数

参数	类型	是否必选	描述
request	<a href="#">PostSignatureRequest</a>	是	<b>参数解释:</b> 基于表单上传请求参数, 详见 <a href="#">PostSignatureRequest</a> 。

表 9-208 PostSignatureRequest

参数	类型	是否必选	描述
bucketName	String	可选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下: <ul style="list-style-type: none"> <li>3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻(如: “my..bucket” )。</li> <li>禁止“.”和“-”相邻(如: “my-.bucket”和“my.-bucket” )。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

参数	类型	是否必选	描述
objectKey	String	可选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
requestDate	Date	可选	<p><b>参数解释:</b> 发起请求的时间。</p> <p><b>默认取值:</b> 无</p>
expiryDate	Date	可选	<p><b>参数解释:</b> 有效截止日期。</p> <p><b>默认取值:</b> 无</p>
expires	long	可选	<p><b>参数解释:</b> 表单上传鉴权的过期时间。</p> <p><b>取值范围:</b> 大于0的整型数。单位：秒。</p> <p><b>默认取值:</b> 300</p>
conditions	List<String>	可选	<p><b>参数解释:</b> 设置表单限制条件，如果设置了该值，将直接使用该值计算policy而忽略请求的表单参数中的设置。</p> <p><b>默认取值:</b> 无</p>
formParams	Map<String, Object>	可选	<p><b>参数解释:</b> 请求的表单参数。参数列表Map中String为请求时的表单参数名称，Object为对应表单参数名称的取值。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 9-209 PostSignatureResponse

参数名称	参数类型	描述
OriginPolicy	String	<b>参数解释:</b> Policy 未经过 Base64 之前的值, 仅用于校验。示例如下: {"expiration":"2023-09-12T12:52:59Z","conditions":[{"content-type":"text/plain"}, {"bucket":"examplebucket"}, {"key":"example/objectname"}]}" <b>默认取值:</b> 无
Policy	String	<b>参数解释:</b> 表单中的 policy, 已经 Base64 之后的值。示例如下: eyJleHBpcmF0aW9uljoiMjAyMy0wOS0xMlQxMjo1Mjo1OVoiLCJjb25kaXRpb25zljpbeyJjb250ZW50LXR5cGUiOiJ0ZXh0L3BsYWlun0seyJidWNrZXQiOiJleGFtcGxlYnVja2V0In0seyJrZXkiOiJleGFtcGxlL29iamVjdG5hbWUifSxdfQ== <b>默认取值:</b> 无
Signature	String	<b>参数解释:</b> 表单中的 signature。示例如下: g0jQr4v9VWd1Q2FOFDG6LGfv9Cw= <b>默认取值:</b> 无

## 代码示例

此用例用于生成带授权信息的表单上传参数 policy 和 signature。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.PostSignatureRequest;
import com.obs.services.model.PostSignatureResponse;
import java.util.HashMap;
import java.util.Map;
public class PostObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK, 也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
// String securityToken = System.getenv("SECURITY_TOKEN");
// endpoint填写桶所在的endpoint, 此处以华北-北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk,endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 生成基于表单上传的请求
    PostSignatureRequest request = new PostSignatureRequest();
    // 设置表单参数
    Map<String, Object> formParams = new HashMap<String, Object>();
    // 设置对象访问权限为公共读
    formParams.put("x-obs-acl", "public-read");
    // 设置对象MIME类型
    formParams.put("content-type", "text/plain");
    request.setFormParams(formParams);
    // 设置表单上传请求有效期，单位：秒
    request.setExpires(3600);
    PostSignatureResponse response = obsClient.createPostSignature(request);
    System.out.println("createPostSignature successfully");
    // 获取表单上传请求参数
    System.out.println("Policy:" + response.getPolicy());
    System.out.println("Signature:" + response.getSignature());
} catch (ObsException e) {
    System.out.println("createPostSignature failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("createPostSignature failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

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>
```



```
<!-- 直接使用PostSignatureResponse.getPolicy()返回的值 -->
<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 ***"/>
<!-- 如果有x-obs-security-token就取消下面这行的注释，并且把x-obs-security-token设置到下面这行的
value里面 -->
<!-- <input type="hidden" name="x-obs-security-token" value="*** Provide your x-obs-security-token
***"/>-->
<input name="file" type="file" />
<input name="submit" value="Upload" type="submit" />
</form>
</body>
</html>
```

## 相关链接

- 关于上传对象-POST上传的API说明，请参见[POST上传](#)。
- 更多关于上传对象的示例代码，请参见[Github示例](#)。
- 上传对象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](#)。
- 上传对象常见问题请参见[上传对象失败](#)。

# 10 下载对象(Java SDK)

## 10.1 对象下载简介(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

OBS Java 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：

- [流式下载](#)
- [范围下载](#)
- [断点续传下载](#)

### 请求参数

参数名	类型	约束	说明
isEncodeHeaders	boolean	可选	是否自动编码请求头，默认是 true。
userHeaders	HashMap<String, String>	可选	用户头域。
ifMatchTag	String	可选	如果对象的ETag值与该参数值相同，则返回对象内容，否则抛出异常。
ifNoneMatchTag	String	可选	如果对象的ETag值与该参数值不相同，则返回对象内容，否则抛出异常。

参数名	类型	约束	说明
ifModifiedSince	Date	可选	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，否则抛出异常。
ifUnmodifiedSince	Date	可选	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，否则抛出异常。
imageProcess	String	可选	<a href="#">图片处理</a> 参数。
rangeStart	Long	可选	范围下载时，指定开始字节。
rangeEnd	Long	可选	范围下载时，指定结束字节。
replaceMetadata	ObjectReplaceMetadata	可选	下载对象时 <a href="#">重写响应头</a> 。
sseCHeader	SseCHeader	可选	服务端加密头信息。
versionId	String	可选	版本号。

## 10.2 流式下载(Java SDK)

### 功能说明

您可以根据需要将存储在OBS中的对象通过ObsClient.getObject接口进行流式下载。ObsClient.getObject返回的ObsObject实例包含对象所在的桶、对象名、对象属性、对象输入流等内容，同时可以通过操作对象输入流将对象的内容读取到本地文件或者内存中。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及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。

- `ObsObject.getObjectContent`获取的对象输入流一定要显式关闭，否则会造成资源泄露。

## 方法定义

`obsClient.getObject(GetObjectRequest request)`

## 请求参数说明

表 10-1 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">GetObjectRequest</a>	必选	<b>参数解释：</b> 桶下载对象时请求参数，详见 <a href="#">GetObjectRequest</a> 。

表 10-2 GetObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释：</b> 桶名。 <b>约束限制：</b> <ul style="list-style-type: none"><li>• 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。</li><li>• 桶命名规则如下：<ul style="list-style-type: none"><li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>- 禁止使用IP地址。</li><li>- 禁止以“-”或“.”开头及结尾。</li><li>- 禁止两个“.”相邻（如：“my..bucket”）。</li><li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>• 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。</li></ul> <b>默认取值：</b> 无

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
imageProcess	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 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<a href="#">处理图片</a>。</p> <p><b>默认取值:</b> 如果不输入处理命令，将返回原图。</p>
rangeStart	Long	可选	<p><b>参数解释:</b> 指定下载对象的起始位置。</p> <p><b>取值范围:</b> 非负整数。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
rangeEnd	Long	可选	<b>参数解释:</b> 指定下载对象的结束位置。 <b>约束限制:</b> 如果该值大于对象长度-1, 实际仍取对象长度-1。 <b>默认取值:</b> 无
ifMatchTag	String	可选	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
ifNoneMatchTag	String	可选	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
IfModifiedSince	Date	可选	<b>参数解释:</b> 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, 否则返回错误。 <b>默认取值:</b> 无
IfUnmodifiedSince	Date	可选	<b>参数解释:</b> 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, 否则返回错误。 <b>默认取值:</b> 无
sseCHeader	<a href="#">SseCHeader</a>	可选	<b>参数解释:</b> 服务端解密头信息, 详见 <a href="#">SseCHeader</a> 。

参数名称	参数类型	是否必选	描述
replaceMetadata	<a href="#">ObjectReplaceMetadata</a>	可选	<b>参数解释:</b> 指定OBS返回请求的相关信息, 详见 <a href="#">ObjectReplaceMetadata</a> 。
progressListener	<a href="#">ProgressListener</a>	可选	<b>参数解释:</b> 设置数据传输监听器, 用于获取下载进度, 详见 <a href="#">ProgressListener</a> 。
encodeHeaders	boolean	可选	<b>参数解释:</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制, 无法发送非ASCII码字符, SDK会在发送请求时对您头域中的 <b>中文汉字</b> 进行url编码, 发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”, 则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。 <b>取值范围:</b> true: 启用SDK编码。 false: 不启用SDK编码。 <b>默认取值:</b> true

表 10-3 ObjectReplaceMetadata

参数名称	参数类型	是否必选	描述
contentType	String	可选	<b>参数解释:</b> 对象的文件类型 (MIME类型)。 contentType (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。 <b>取值范围:</b> 常见的contentType (MIME) 列表参见 <a href="#">如何理解Content-Type (MIME)? (Java SDK)</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentLanguage	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>
expires	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示或在访问时弹出文件下载对话框。</p> <p><b>默认取值:</b> 无</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。</p> <p><b>默认取值:</b> 无</p>



表 10-4 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释:</b> 获取上传进度, 详见 <a href="#">progressChanged</a> 。 <b>默认取值:</b> 无

表 10-5 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释:</b> 传输进度数据, 详见 <a href="#">ProgressStatus</a> 。 <b>默认取值:</b> 无

表 10-6 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 10-7 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard，AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥，byte[]形式，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥，由原始密钥经过Base64编码后得到，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 10-8 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 10-9 ServerAlgorithm

常量值	原始值
AES256	AES256

表 10-10 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 10-11 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 10-12 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<b>参数解释:</b> 对象数据的长度。 <b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。
contentType	String	可选	<b>参数解释:</b> 对象的文件类型 ( MIME类型 )。 contentType ( 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。 <b>取值范围:</b> 常见的contentType ( MIME ) 列表参见 <a href="#">如何理解Content-Type ( MIME ) ? (Java SDK)</a> 。 <b>默认取值:</b> 如果未指定Content-Type, SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type ( 如.xml判断为application/xml文件; .html判断为text/html文件 )。
contentEncoding	String	可选	<b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。 <b>默认取值:</b> 无
contentDisposition	String	可选	<b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象, 当下载对象或者访问对象时, 以默认文件名命名的文件将直接在浏览器上显示, 或在弹出文件下载对话框时显示。 <b>默认取值:</b> 无
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentLanguage	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无
contentMd5	String	可选	<b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配，返回HTTP 400错误。 <b>约束限制:</b> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <b>取值范围:</b> 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== <b>默认取值:</b> 无
storageClasses	<a href="#">StorageClassEnum</a>	可选	<b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。 <b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b></p> <p>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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li><li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li></ul> <p><b>默认取值:</b></p> <p>无</p>

## 返回结果说明

表 10-13 ObsObject

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
owner	<b>Owner</b>	<p><b>参数解释:</b> 所有者，详见<b>Owner</b>。</p>
metadata	<b>ObjectMetadata</b>	<p><b>参数解释:</b> 对象元数据，详见<b>ObjectMetadata</b>。</p>



参数名称	参数类型	描述
objectContent	InputStream	<b>参数解释:</b> 对象数据流。 <b>默认取值:</b> 无

## 代码示例

本示例用于流式下载examplebucket桶中的objectname对象。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObsObject;
import java.io.ByteArrayOutputStream;
import java.io.InputStream;
public class GetObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。查看桶所在的endpoint请参见：https://support.huaweicloud.com/usermanual-obs/obs\_03\_0312.html。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 流式下载
            ObsObject obsObject = obsClient.getObject("examplebucket", "objectname");
            // 读取对象内容
            System.out.println("Object content:");
            InputStream input = obsObject.getObjectContent();
            byte[] b = new byte[1024];
            ByteArrayOutputStream bos = new ByteArrayOutputStream();
            int len;
            while ((len = input.read(b)) != -1) {
                bos.write(b, 0, len);
            }
            System.out.println("getObjectContent successfully");
            System.out.println(new String(bos.toByteArray()));
            bos.close();
            input.close();
        } catch (ObsException e) {
            System.out.println("getObjectContent failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
        }
    }
}
```

```
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("getObjectContent failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于下载对象的API说明, 请参见[获取对象内容](#)。
- 更多关于下载对象的示例代码, 请参见[Github示例](#)。
- 下载对象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](#)。
- 下载对象常见问题请参见[下载对象失败](#)。

## 10.3 范围下载(Java SDK)

### 功能说明

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1到第1001个字节的数据, 共1001字节的数据。如果指定的范围无效, 则返回整个对象的数据。

#### 须知

开发过程中, 您有任何问题可以在github上[提交issue](#), 或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及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, 开始位置大于结束位置), 接口会抛出异常, 如果指定范围有效且结束位置大于文件大小, 则返回整个对象。

### 方法定义

obsClient.getObject([GetObjectRequest request](#))

## 请求参数说明

表 10-14 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">GetObjectRequest</a>	必选	<b>参数解释:</b> 桶下载对象时请求参数, 详见 <a href="#">GetObjectRequest</a> 。

表 10-15 GetObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下: <ul style="list-style-type: none"> <li>3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻(如: “my..bucket” )。</li> <li>禁止“.”和“-”相邻(如: “my-.bucket”和“my-bucket” )。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
imageProcess	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 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<a href="#">处理图片</a>。</p> <p><b>默认取值:</b> 如果不输入处理命令，将返回原图。</p>
rangeStart	Long	可选	<p><b>参数解释:</b> 指定下载对象的起始位置。</p> <p><b>取值范围:</b> 非负整数。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
rangeEnd	Long	可选	<p><b>参数解释:</b> 指定下载对象的结束位置。</p> <p><b>约束限制:</b> 如果该值大于对象长度-1，实际仍取对象长度-1。</p> <p><b>默认取值:</b> 无</p>
ifMatchTag	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
ifNoneMatchTag	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
IfModifiedSince	Date	可选	<p><b>参数解释:</b> 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，否则返回错误。</p> <p><b>默认取值:</b> 无</p>
IfUnmodifiedSince	Date	可选	<p><b>参数解释:</b> 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，否则返回错误。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端解密头信息，详见<a href="#">SseCHeader</a>。</p>
replaceMetadata	<a href="#">ObjectReplaceMetadata</a>	可选	<p><b>参数解释:</b> 指定OBS返回请求的相关信息，详见<a href="#">ObjectReplaceMetadata</a>。</p>

参数名称	参数类型	是否必选	描述
progressListener	<a href="#">ProgressListener</a>	可选	<b>参数解释:</b> 设置数据传输监听器，用于获取下载进度，详见 <a href="#">ProgressListener</a> 。
encodeHeaders	boolean	可选	<b>参数解释:</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制，无法发送非ASCII码字符，SDK会在发送请求时对您头域中的中文汉字进行url编码，发送编码后数据。如您设置的值content-disposition为“attachment; filename=“中文.txt””，则对象元数据中存储的信息为“attachment; filename=%E4%B8%AD%E6%96%87.txt”。使用浏览器访问时浏览器将会自动解码。 <b>取值范围:</b> true: 启用SDK编码。 false: 不启用SDK编码。 <b>默认取值:</b> true

表 10-16 ObjectRepleaceMetadata

参数名称	参数类型	是否必选	描述
contentType	String	可选	<b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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。 <b>取值范围:</b> 常见的contentType（MIME）列表参见 <a href="#">如何理解Content-Type（MIME）？(Java SDK)</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>
expires	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示或在访问时弹出文件下载对话框。</p> <p><b>默认取值:</b> 无</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。</p> <p><b>默认取值:</b> 无</p>

表 10-17 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释:</b> 获取上传进度, 详见 <a href="#">progressChanged</a> 。 <b>默认取值:</b> 无

表 10-18 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释:</b> 传输进度数据, 详见 <a href="#">ProgressStatus</a> 。 <b>默认取值:</b> 无

表 10-19 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。



表 10-20 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard, AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥，byte[]形式，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥，由原始密钥经过Base64编码后得到，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 10-21 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 10-22 ServerAlgorithm

常量值	原始值
AES256	AES256

表 10-23 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 10-24 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 10-25 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型 ( MIME类型 )。contentType ( 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><b>取值范围:</b> 常见的contentType ( MIME ) 列表参见<a href="#">如何理解Content-Type ( MIME ) ? (Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type, SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type ( 如.xml判断为application/xml文件; .html判断为text/html文件 )。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象, 当下载对象或者访问对象时, 以默认文件名命名的文件将直接在浏览器上显示, 或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>
expires	String	可选	<p><b>参数解释:</b> 指对象在网页中的缓存过期时间。</p> <p><b>约束限制:</b> 日期格式为GMT的格式。</p> <p><b>默认取值:</b> 无</p>
contentTypeMD5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>对象数据的MD5值必须经过Base64编码。</li> <li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li> </ul> <p><b>取值范围:</b> 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：n58lG6hfM7vql4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>
storageClass	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li><li>• 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。</li></ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 10-26 ObsObject

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 桶命名规则如下： <ul style="list-style-type: none"> <li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>- 禁止使用IP地址。</li> <li>- 禁止以“-”或“.”开头及结尾。</li> <li>- 禁止两个“.”相邻（如：“my..bucket”）。</li> <li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
owner	Owner	<p><b>参数解释:</b> 所有者，详见Owner。</p>
metadata	ObjectMetadata	<p><b>参数解释:</b> 对象元数据，详见ObjectMetadata。</p>
objectContent	InputStream	<p><b>参数解释:</b> 对象数据流。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于范围下载examplebucket桶中的objectname对象，并且指定下载范围是0L~1000L。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.GetObjectRequest;
import com.obs.services.model.ObsObject;
import java.io.InputStream;
public class GetObject002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
```

```
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 范围下载
    GetObjectRequest request = new GetObjectRequest("examplebucket", "objectname");
    // 指定开始和结束范围
    request.setRangeStart(0);
    request.setRangeEnd(1000);
    ObsObject obsObject = obsClient.getObject(request);
    // 读取数据
    byte[] buf = new byte[1024];
    InputStream in = obsObject.getObjectContent();
    for (int n = 0; n != -1; ) {
        n = in.read(buf, 0, buf.length);
    }
    System.out.println("GetObject successfully");
    in.close();
} catch (ObsException e) {
    System.out.println("GetObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("GetObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于下载对象的API说明，请参见[获取对象内容](#)。
- 更多关于下载对象的示例代码，请参见[Github示例](#)。
- 下载对象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](#)。
- 下载对象常见问题请参见[下载对象失败](#)。
- 可以利用范围下载并发下载大对象，详细代码示例请参考[ConcurrentDownloadObjectSample](#)。

## 10.4 获取下载进度(Java SDK)

### 功能说明

获取指定对象的下载进度。

您可以通过GetObjectRequest.setProgressInterval设置数据传输接口来获取下载的进度。



**须知**

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

**接口约束**

- 您必须是桶拥有者或拥有下载对象的权限，才能获取获取下载进度。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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 支持获取下载进度的接口包括：流式下载、范围下载和断点续传下载。

**方法定义**

GetObjectRequest.setProgressListener([ProgressListener progressListener](#))

**参数说明**

表 10-27 请求参数列表

参数名称	参数类型	是否必选	描述
progressListener	<a href="#">ProgressListener</a>	必选	<b>参数解释：</b> 设置数据传输监听器，用于获取下载进度，详见 <a href="#">ProgressListener</a> 。

表 10-28 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释：</b> 获取上传进度，详见 <a href="#">progressChanged</a> 。

表 10-29 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释：</b> 传输进度数据，详见 <a href="#">ProgressStatus</a> 。

表 10-30 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

## 代码示例

本示例用于获取下载examplebucket桶中objectname对象时的进度。示例代码如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.GetObjectRequest;
import com.obs.services.model.ObsObject;
import com.obs.services.model.ProgressListener;
import com.obs.services.model.ProgressStatus;
import java.io.ByteArrayOutputStream;
import java.io.InputStream;
public class GetObject003 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 获取下载进度
            GetObjectRequest request = new GetObjectRequest("examplebucket", "objectname");
            request.setProgressListener(
                new ProgressListener() {
                    @Override
                    public void progressChanged(ProgressStatus status) {
                        // 获取下载平均速率

```

```
        System.out.println("AverageSpeed:" + status.getAverageSpeed());
        // 获取下载进度百分比
        System.out.println("TransferPercentage:" + status.getTransferPercentage());
    }
    });
    // 每下载1MB数据反馈下载进度
    request.setProgressInterval(1024 * 1024L);
    ObsObject obsObject = obsClient.getObject(request);
    // 读取对象内容
    System.out.println("Object content:");
    InputStream input = obsObject.getObjectContent();
    byte[] b = new byte[1024];
    ByteArrayOutputStream bos = new ByteArrayOutputStream();
    int len;
    while ((len = input.read(b)) != -1) {
        bos.write(b, 0, len);
    }
    System.out.println("getObjectContent successfully");
    System.out.println(new String(bos.toByteArray()));
    bos.close();
    input.close();
} catch (ObsException e) {
    System.out.println("getObjectContent failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getObjectContent failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于下载对象的API说明，请参见[获取对象内容](#)。
- 更多关于下载对象的示例代码，请参见[Github示例](#)。
- 下载对象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](#)。
- 下载对象常见问题请参见[下载对象失败](#)。

## 10.5 限定条件下下载(Java SDK)

### 功能说明

下载对象时，可以指定一个或多个限定条件，满足限定条件时则进行下载，否则返回异常码，下载对象失败。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 接口约束

- 您必须是桶所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以及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([GetObjectRequest request](#))

## 请求参数说明

表 10-31 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">GetObjectRequest</a>	必选	<b>参数解释：</b> 获取桶下载对象的请求参数，详见 <a href="#">GetObjectRequest</a> 。

表 10-32 GetObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my.bucket”和“my.bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
imageProcess	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 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<a href="#">处理图片</a>。</p> <p><b>默认取值:</b> 如果不输入处理命令, 将返回原图。</p>
rangeStart	Long	可选	<p><b>参数解释:</b> 指定下载对象的起始位置。</p> <p><b>取值范围:</b> 非负整数。</p> <p><b>默认取值:</b> 无</p>
rangeEnd	Long	可选	<p><b>参数解释:</b> 指定下载对象的结束位置。</p> <p><b>约束限制:</b> 如果该值大于对象长度-1, 实际仍取对象长度-1。</p> <p><b>默认取值:</b> 无</p>
ifMatchTag	String	可选	<p><b>参数解释:</b></p> <ol style="list-style-type: none"> <li>对象的ETag值即对象数据的MD5校验值。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。</li> <li>如果包含If-Unmodified-Since并且不符合或者包含If-Match并且不符合, 抛出异常中HTTP状态码为: 412 precondition failed。</li> </ol> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
ifNoneMatchTag	String	可选	<p><b>参数解释:</b></p> <ol style="list-style-type: none"> <li>1. 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。</li> <li>2. 如果包含If-Modified-Since并且不符合或者包含If-None-Match并且不符合, 抛出异常中HTTP状态码为: 304 Not Modified。</li> </ol> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
IfModifiedSince	Date	可选	<p><b>参数解释:</b></p> <ol style="list-style-type: none"> <li>1. 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, 否则返回错误。</li> <li>2. 如果包含If-Modified-Since并且不符合或者包含If-None-Match并且不符合, 抛出异常中HTTP状态码为: 304 Not Modified。</li> </ol> <p><b>默认取值:</b> 无</p>
IfUnmodifiedSince	Date	可选	<p><b>参数解释:</b></p> <ol style="list-style-type: none"> <li>1. 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, 否则返回错误。</li> <li>2. 如果包含If-Unmodified-Since并且不符合或者包含If-Match并且不符合, 抛出异常中HTTP状态码为: 412 precondition failed。</li> </ol> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端解密头信息, 详见<a href="#">SseCHeader</a>。</p>
replaceMetadata	<a href="#">ObjectReplaceMetadata</a>	可选	<p><b>参数解释:</b> 指定OBS返回请求的相关信息, 详见<a href="#">ObjectReplaceMetadata</a>。</p>
progressListener	<a href="#">ProgressListener</a>	可选	<p><b>参数解释:</b> 设置数据传输监听器, 用于获取下载进度, 详见<a href="#">ProgressListener</a>。</p>

参数名称	参数类型	是否必选	描述
encodeHeaders	boolean	可选	<p><b>参数解释：</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制，无法发送非ASCII码字符，SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码，发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”，则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围：</b> true：启用SDK编码。 false：不启用SDK编码。</p> <p><b>默认取值：</b> true</p>

表 10-33 ObjectRepaceMetadata

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释：</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围：</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值：</b> 无</p>
contentType	String	可选	<p><b>参数解释：</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值：</b> 无</p>



参数名称	参数类型	是否必选	描述
expires	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示或在访问时弹出文件下载对话框。</p> <p><b>默认取值:</b> 无</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。</p> <p><b>默认取值:</b> 无</p>

表 10-34 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<p><b>参数解释:</b> 获取上传进度，详见<a href="#">progressChanged</a>。</p> <p><b>默认取值:</b> 无</p>

表 10-35 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<p><b>参数解释:</b> 传输进度数据, 详见 <a href="#">ProgressStatus</a>。</p> <p><b>默认取值:</b> 无</p>

表 10-36 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 10-37 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式, 对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值: AES256, 即选择SSE-C方式, 使用高级加密标准 (Advanced Encryption Standard, AES) 加密对象。详见 <a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<b>参数解释:</b> 加密算法。 <b>约束限制:</b> 只支持AES256。 <b>取值范围:</b> 详见 <a href="#">SSEAlgorithmEnum</a> 。 <b>默认取值:</b> 无
sseCKey	byte[]	必选	<b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。 <b>默认取值:</b> 无
sseCKeyBase64	String	可选	<b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过Base64编码后得到, 该密钥用于加密对象。 <b>默认取值:</b> 无

表 10-38 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 10-39 ServerAlgorithm

常量值	原始值
AES256	AES256

表 10-40 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 10-41 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 10-42 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。</p>

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型 ( MIME类型 )。contentType ( 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><b>取值范围:</b> 常见的contentType ( MIME ) 列表参见<a href="#">如何理解Content-Type ( MIME ) ? (Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type, SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type ( 如.xml判断为application/xml文件; .html判断为text/html文件 )。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象, 当下载对象或者访问对象时, 以默认文件名命名的文件将直接在浏览器上显示, 或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentTypeLanguage	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合, 以自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
expires	String	可选	<p><b>参数解释:</b> 指对象在网页中的缓存过期时间。</p> <p><b>约束限制:</b> 日期格式为GMT的格式。</p> <p><b>默认取值:</b> 无</p>
contentMd5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <p><b>取值范围:</b> 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：n58lG6hfM7vql4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>
storageClasses	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 10-43 ObsObject

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 桶命名规则如下： <ul style="list-style-type: none"> <li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>- 禁止使用IP地址。</li> <li>- 禁止以“-”或“.”开头及结尾。</li> <li>- 禁止两个“.”相邻（如：“my..bucket”）。</li> <li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.bucket”）。</li> </ul> </li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	描述
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
owner	Owner	<p><b>参数解释:</b> 所有者，详见Owner。</p>
metadata	ObjectMetadata	<p><b>参数解释:</b> 对象元数据，详见ObjectMetadata。</p>
objectContent	InputStream	<p><b>参数解释:</b> 对象数据流。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于限定条件下下载examplebucket桶中的objectname对象，并且指定该文件的修改时间必须在“2016-01-01”之后才能下载成功。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.GetObjectRequest;
import com.obs.services.model.ObsObject;
import java.text.SimpleDateFormat;
public class GetObject004 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
```

```
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 限定条件下载
    GetObjectRequest request = new GetObjectRequest("examplebucket", "objectname");
    request.setRangeStart(0L);
    request.setRangeEnd(1000L);
    request.setIfModifiedSince(new SimpleDateFormat("yyyy-MM-dd").parse("2016-01-01"));
    ObsObject obsObject = obsClient.getObject(request);
    System.out.println("getObject successfully");
    obsObject.getObjectContent().close();
} catch (ObsException e) {
    System.out.println("getObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于下载对象的API说明,请参见[获取对象内容](#)。
- 更多关于下载对象的示例代码,请参见[Github示例](#)。
- 下载对象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](#)。
- 下载对象常见问题请参见[下载对象失败](#)。

## 10.6 重写响应头(Java SDK)

### 功能说明

下载对象时,可以重写HTTP/HTTPS中部分响应头的信息:Content-Type、Content-Language、Expires、Cache-Control、Content-Disposition、Content-Encoding。

#### 须知

开发过程中,您有任何问题可以在github上[提交issue](#),或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及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.getObject([GetObjectRequest request](#))

## 请求参数说明

表 10-44 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">GetObjectRequest</a>	必选	<b>参数解释:</b> 获取桶下载对象的请求参数，详见 <a href="#">GetObjectRequest</a> 。

表 10-45 GetObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 桶命名规则如下： <ul style="list-style-type: none"> <li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>- 禁止使用IP地址。</li> <li>- 禁止以“-”或“.”开头及结尾。</li> <li>- 禁止两个“.”相邻（如：“my.bucket”）。</li> <li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
imageProcess	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 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<a href="#">处理图片</a>。</p> <p><b>默认取值:</b> 如果不输入处理命令，将返回原图。</p>
rangeStart	Long	可选	<p><b>参数解释:</b> 指定下载对象的起始位置。</p> <p><b>取值范围:</b> 非负整数。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
rangeEnd	Long	可选	<p><b>参数解释:</b> 指定下载对象的结束位置。</p> <p><b>约束限制:</b> 如果该值大于对象长度-1，实际仍取对象长度-1。</p> <p><b>默认取值:</b> 无</p>
ifMatchTag	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
ifNoneMatchTag	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
IfModifiedSince	Date	可选	<p><b>参数解释:</b> 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，否则返回错误。</p> <p><b>默认取值:</b> 无</p>
IfUnmodifiedSince	Date	可选	<p><b>参数解释:</b> 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，否则返回错误。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<b>SseCHeader</b>	可选	<p><b>参数解释:</b> 服务端解密头信息，详见<b>SseCHeader</b>。</p>

参数名称	参数类型	是否必选	描述
replaceMetadata	<a href="#">ObjectReplaceMetadata</a>	可选	<b>参数解释:</b> 指定OBS返回请求的相关信息, 详见 <a href="#">ObjectReplaceMetadata</a> 。
progressListener	<a href="#">ProgressListener</a>	可选	<b>参数解释:</b> 设置数据传输监听器, 用于获取下载进度, 详见 <a href="#">ProgressListener</a> 。

表 10-46 ObjectReplaceMetadata

参数名称	参数类型	是否必选	描述
contentType	String	可选	<b>参数解释:</b> 对象的文件类型 ( MIME类型 ) 。 contentType ( 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。 <b>取值范围:</b> 常见的contentType ( MIME ) 列表参见 <a href="#">如何理解Content-Type ( MIME ) ? (Java SDK)</a> 。 <b>默认取值:</b> 无
contentType	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合, 以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无
expires	String	可选	<b>参数解释:</b> 响应中的Expires头。指定对象被下载时的网页的缓存过期时间。 <b>默认取值:</b> 无
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头, 指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示或在访问时弹出文件下载对话框。</p> <p><b>默认取值:</b> 无</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。</p> <p><b>默认取值:</b> 无</p>

表 10-47 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<p><b>参数解释:</b> 获取上传进度，详见<a href="#">progressChanged</a></p> <p><b>默认取值:</b> 无</p>

表 10-48 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<p><b>参数解释:</b> 传输进度数据，详见<a href="#">ProgressStatus</a></p> <p><b>默认取值:</b> 无</p>

表 10-49 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 10-50 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard, AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
sseCKey	byte[]	必选	<b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。 <b>默认取值:</b> 无
sseCKeyBase64	String	可选	<b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过Base64编码后得到, 该密钥用于加密对象。 <b>默认取值:</b> 无

表 10-51 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 10-52 ServerAlgorithm

常量值	原始值
AES256	AES256

表 10-53 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 10-54 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 10-55 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型 ( MIME类型 )。contentType ( 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><b>取值范围:</b> 常见的contentType ( MIME ) 列表参见<a href="#">如何理解Content-Type ( MIME ) ? (Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type, SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type ( 如.xml判断为application/xml文件; .html判断为text/html文件 )。</p>

参数名称	参数类型	是否必选	描述
contentEncoding	String	可选	<b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。 <b>默认取值:</b> 无
contentDisposition	String	可选	<b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象,当下载对象或者访问对象时,以默认文件名命名的文件将直接在浏览器上显示,或在弹出文件下载对话框时显示。 <b>默认取值:</b> 无
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无
contentType	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合,以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentMd5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <p><b>取值范围:</b> 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：n58IG6hfM7vqI4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>
storageClass	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b></p> <p>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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li><li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li></ul> <p><b>默认取值:</b></p> <p>无</p>

## 返回结果说明

表 10-56 ObsObject

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
owner	<b>Owner</b>	<p><b>参数解释:</b> 所有者，详见<b>Owner</b>。</p>
metadata	<b>ObjectMetadata</b>	<p><b>参数解释:</b> 对象元数据，详见<b>ObjectMetadata</b>。</p>

参数名称	参数类型	描述
objectContent	InputStream	<b>参数解释:</b> 对象数据流。 <b>默认取值:</b> 无

## 代码示例

以下代码展示了如何重写响应头，并下载examplebucket桶中的objectname对象。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.GetObjectRequest;
import com.obs.services.model.ObjectRepleaceMetadata;
import com.obs.services.model.ObsObject;
public class GetObject005 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 重写响应头
            GetObjectRequest request = new GetObjectRequest("examplebucket", "objectname");
            ObjectRepleaceMetadata replaceMetadata = new ObjectRepleaceMetadata();
            replaceMetadata.setContentType("image/jpeg");
            request.setReplaceMetadata(replaceMetadata);
            ObsObject obsObject = obsClient.getObject(request);
            System.out.println(obsObject.getMetadata().getContentType());
            System.out.println("getObject successfully");
            obsObject.getObjectContent().close();
        } catch (ObsException e) {
            System.out.println("getObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("getObject failed");
            // 其他异常信息打印
        }
    }
}
```



```
e.printStackTrace();  
    }  
}  
}
```

## 相关链接

- 关于下载对象的API说明，请参见[获取对象内容](#)。
- 更多关于下载对象的示例代码，请参见[Github示例](#)。
- 下载对象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](#)。
- 下载对象常见问题请参见[下载对象失败](#)。

## 10.7 获取自定义元数据(Java SDK)

### 功能说明

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

本接口可以在下载对象成功后返回对象的自定义元数据。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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，才能获取自定义元数据。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以及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.getObject([GetObjectRequest request](#))

### 请求参数说明

表 10-57 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">GetObjectRequest</a>	必选	<b>参数解释：</b> 桶下载对象时请求参数，详见 <a href="#">GetObjectRequest</a> 。

表 10-58 GetObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
imageProcess	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 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<a href="#">处理图片</a>。</p> <p><b>默认取值:</b> 如果不输入处理命令，将返回原图。</p>
rangeStart	Long	可选	<p><b>参数解释:</b> 指定下载对象的起始位置。</p> <p><b>取值范围:</b> 非负整数。</p> <p><b>默认取值:</b> 无</p>
rangeEnd	Long	可选	<p><b>参数解释:</b> 指定下载对象的结束位置。</p> <p><b>约束限制:</b> 如果该值大于对象长度-1，实际仍取对象长度-1。</p> <p><b>默认取值:</b> 无</p>
ifMatchTag	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
ifNoneMatchTag	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
IfModifiedSince	Date	可选	<p><b>参数解释:</b> 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，否则返回错误。</p> <p><b>默认取值:</b> 无</p>
IfUnmodifiedSince	Date	可选	<p><b>参数解释:</b> 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，否则返回错误。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端解密头信息，详见<a href="#">SseCHeader</a>。</p>
replaceMetadata	<a href="#">ObjectReplaceMetadata</a>	可选	<p><b>参数解释:</b> 指定OBS返回请求的相关信息，详见<a href="#">ObjectReplaceMetadata</a>。</p>
progressListener	<a href="#">ProgressListener</a>	可选	<p><b>参数解释:</b> 设置数据传输监听器，用于获取下载进度，详见<a href="#">ProgressListener</a>。</p>

参数名称	参数类型	是否必选	描述
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制，无法发送非ASCII码字符，SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码，发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”，则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围:</b> true：启用SDK编码。 false：不启用SDK编码。</p> <p><b>默认取值:</b> true</p>

表 10-59 ObjectRepleaceMetadata

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 无</p>
contentType	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
expires	String	可选	<b>参数解释:</b> 响应中的Expires头。指定对象被下载时的网页的缓存过期时间。 <b>默认取值:</b> 无
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头, 指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无
contentDisposition	String	可选	<b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象, 当下载对象或者访问对象时, 以默认文件名命名的文件将直接在浏览器上显示或在访问时弹出文件下载对话框。 <b>默认取值:</b> 无
contentEncoding	String	可选	<b>参数解释:</b> 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。 <b>默认取值:</b> 无

表 10-60 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释:</b> 获取上传进度, 详见 <a href="#">progressChanged</a> <b>默认取值:</b> 无

表 10-61 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释:</b> 传输进度数据, 详见 <a href="#">ProgressStatus</a> <b>默认取值:</b> 无

表 10-62 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 10-63 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<b>参数解释:</b> 表示服务端加密是SSE-C方式, 对象使用SSE-C加密方式。 <b>取值范围:</b> 可选值: AES256, 即选择SSE-C方式, 使用高级加密标准 (Advanced Encryption Standard, AES) 加密对象。详见 <a href="#">ServerAlgorithm</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<b>参数解释:</b> 加密算法。 <b>约束限制:</b> 只支持AES256。 <b>取值范围:</b> 详见 <a href="#">SSEAlgorithmEnum</a> 。 <b>默认取值:</b> 无
sseCKey	byte[]	必选	<b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。 <b>默认取值:</b> 无
sseCKeyBase64	String	可选	<b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过Base64编码后得到, 该密钥用于加密对象。 <b>默认取值:</b> 无

表 10-64 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 10-65 ServerAlgorithm

常量值	原始值
AES256	AES256



表 10-66 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 10-67 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 10-68 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<b>参数解释:</b> 对象数据的长度。 <b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentTypeLanguage	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无
contentMd5	String	可选	<b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配，返回HTTP 400错误。 <b>约束限制:</b> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <b>取值范围:</b> 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：n58lG6hfM7vql4K0vnWpog== <b>默认取值:</b> 无
storageClasses	<a href="#">StorageClassEnum</a>	可选	<b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。 <b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 10-69 ObsObject

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 桶命名规则如下： <ul style="list-style-type: none"> <li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>- 禁止使用IP地址。</li> <li>- 禁止以“-”或“.”开头及结尾。</li> <li>- 禁止两个“.”相邻（如：“my..bucket”）。</li> <li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
owner	Owner	<p><b>参数解释:</b> 所有者，详见Owner。</p>
metadata	ObjectMetadata	<p><b>参数解释:</b> 对象元数据，详见ObjectMetadata。</p>
objectContent	InputStream	<p><b>参数解释:</b> 对象数据流。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例通过下载examplebucket桶中的objectname对象成功后，返回自定义元数据。示例代码如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.GetObjectRequest;
import com.obs.services.model.ObjectMetadata;
import com.obs.services.model.ObsObject;
import com.obs.services.model.PutObjectRequest;
public class GetObject006 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
```

```
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 获取自定义元数据
    // 上传对象, 设置自定义元数据
    PutObjectRequest request = new PutObjectRequest("examplebucket", "objectname");
    ObjectMetadata metadata = new ObjectMetadata();
    metadata.addUserMetadata("property", "property-value");
    request.setMetadata(metadata);
    obsClient.putObject(request);
    // 下载对象, 获取对象自定义元数据
    GetObjectRequest request1 = new GetObjectRequest("examplebucket", "objectname");
    ObsObject obsObject = obsClient.getObject(request1);
    System.out.println("getObject successfully");
    // 读取对象元数据
    System.out.println(obsObject.getMetadata().getContentType());
    // 获取名为property的自定义元数据
    System.out.println(obsObject.getMetadata().getUserMetadata("property"));
    // 获取所有自定义元数据
    System.out.println("allUserMetadata:" + metadata.getAllMetadata());
    obsObject.getObjectContent().close();
} catch (ObsException e) {
    System.out.println("getObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于获取对象元数据的API说明, 请参见[获取对象元数据](#)。
- 更多关于获取对象元数据的示例代码, 请参见[Github示例](#)。
- 获取对象元数据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](#)。
- 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。

## 10.8 恢复归档存储对象(Java SDK)

### 功能说明

如果要下载归档存储对象, 需要先将归档存储对象恢复。恢复归档存储对象的恢复选项可支持两类, 见下表:

选项	说明	OBS Java SDK对应值
快速恢复	恢复耗时1~5分钟。	RestoreTierEnum.EXPEDITED
标准恢复	恢复耗时3~5小时。默认值。	RestoreTierEnum.STANDARD

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- ObsClient.restoreObject中指定的对象必须是归档存储类别，否则调用该接口会抛出异常。

## 方法定义

obsClient.restoreObject([RestoreObjectRequest request](#))

## 请求参数说明

表 10-70 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">RestoreObjectRequest</a>	必选	<b>参数解释：</b> 桶下载对象时请求参数，详见 <a href="#">RestoreObjectRequest</a> 。



表 10-71 RestoreObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 待恢复归档存储对象的版本号。</p> <p><b>默认取值:</b> 无，如果不设置则默认指定最新版本的对象。</p>

参数名称	参数类型	是否必选	描述
days	int	必选	<p><b>参数解释:</b> 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><b>约束限制:</b> 取值必须是正整数。</p> <p><b>取值范围:</b> [1, 30], 单位: 天。</p> <p><b>默认取值:</b> 无</p>
tier	RestoreTierEnum	可选	<p><b>参数解释:</b> 恢复选项。表示恢复对象所耗的时间。</p> <p><b>取值范围:</b> 详见RestoreTierEnum。</p> <p><b>默认取值:</b> 默认为标准恢复。</p>

表 10-72 RestoreTierEnum

可选值	原始值	描述
EXPEDITED	Expedited	快速恢复, 恢复耗时1~5分钟。
STANDARD	Standard	标准恢复, 恢复耗时3~5小时。

## 返回结果说明

表 10-73 RestoreObjectStatus

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例展示了通过ObsClient.restoreObject恢复归档存储对象。然后通过ObsClient.getObject下载examplebucket桶下的objectname对象。示例代码如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObsObject;
import com.obs.services.model.RestoreObjectRequest;
import com.obs.services.model.RestoreTierEnum;
public class GetObject007 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 下载归档存储对象
            RestoreObjectRequest request = new RestoreObjectRequest();
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname");
            request.setDays(1);
            request.setRestoreTier(RestoreTierEnum.EXPEDITED);
            obsClient.restoreObject(request);
            // 等待对象恢复
            Thread.sleep(60 * 6 * 1000);
            // 下载对象
            ObsObject obsObject = obsClient.getObject("examplebucket", "objectname");
            System.out.println("getObject successfully");
            obsObject.getObjectContent().close();
        } catch (ObsException e) {
            System.out.println("getObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
        }
    }
}
```

```
System.out.println("Error Message:" + e.getMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("getObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于恢复归档存储对象的API说明, 请参见[恢复归档存储对象](#)。
- 更多关于恢复归档存储对象的示例代码, 请参见[Github示例](#)。
- 恢复归档存储对象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](#)。

## 10.9 断点续传下载(Java SDK)

### 功能说明

1. 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, 否则抛出异常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)。
2. 断点续传下载接口是利用[范围下载\(Java SDK\)](#)特性实现的, 是对范围下载的封装和加强。
3. 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。

#### 须知

开发过程中, 您有任何问题可以在github上[提交issue](#), 或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及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.downloadFile([DownloadFileRequest request](#))

## 请求参数说明

表 10-74 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">DownloadFileRequest</a>	必选	<b>参数解释:</b> 下载大对象请求参数列表, 详见 <a href="#">DownloadFileRequest</a> 。

表 10-75 DownloadFileRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释:</b> 桶名。 <b>约束限制:</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下:<ul style="list-style-type: none"><li>3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻(如: “my..bucket”)。</li><li>禁止“.”和“-”相邻(如: “my-.bucket”和“my.-bucket”)。</li></ul></li><li>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。</li></ul> <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无
downloadFile	String	可选	<b>参数解释:</b> 下载对象的本地文件全路径。 <b>默认取值:</b> 当该值为空时，默认为当前程序的运行目录。
partSize	long	可选	<b>参数解释:</b> 分段大小。 <b>取值范围:</b> 取值范围是100KB~5GB，单位：字节。 <b>默认取值:</b> 9MB
taskNum	int	可选	<b>参数解释:</b> 分段下载时的最大并发数。 <b>取值范围:</b> (0, 文件大小/分段大小]，即大于0小于等于文件大小除以分段大小向上取整。 <b>默认取值:</b> 1，即不设置则默认单任务下载。
enableCheckpoint	boolean	可选	<b>参数解释:</b> 是否开启断点续传模式。 <b>取值范围:</b> true：开启断点续传模式。 false：关闭断点续传模式。 <b>默认取值:</b> false

参数名称	参数类型	是否必选	描述
checkpointFile	String	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 仅在断点续传模式下有效。</p> <p><b>默认取值:</b> 当该值为空时，默认为当前目录。</p>
enableChecksum	boolean	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 仅在断点续传模式下有效。</p> <p><b>取值范围:</b> true: 校验待上传文件的内容。 false: 不校验待上传文件的内容。</p> <p><b>默认取值:</b> false</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号，用于获取指定版本号的对象。例如： G001117FCE89978B0000401205D5DC9。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无，如果不设置则默认获取最新版本的对象。</p>
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否自动解码响应头</p> <p><b>默认取值:</b> 无</p>
ifModifiedSince	Date	可选	<p><b>参数解释:</b> 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，否则返回错误。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
ifUnmodified Since	Date	可选	<p><b>参数解释:</b> 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, 否则返回错误。</p> <p><b>默认取值:</b> 无</p>
ifMatchTag	String	可选	<p><b>参数解释:</b> 如果对象的ETag值与该参数值相同, 则返回对象内容, 否则抛出异常。</p> <p><b>默认取值:</b> 无</p>
ifNoneMatch Tag	String	可选	<p><b>参数解释:</b> 如果对象的ETag值与该参数值不相同, 则返回对象内容, 否则抛出异常。</p> <p><b>默认取值:</b> 无</p>
progressListener	<b>ProgressListener</b>	可选	<p><b>参数解释:</b> 设置数据传输监听器, 用于获取下载进度。详见<a href="#">ProgressListener</a>。</p>
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制, 无法发送非ASCII码字符, SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码, 发送编码后数据。如您设置的值content-disposition为“attachment; filename=“中文.txt””, 则对象元数据中存储的信息为“attachment; filename=“%E4%B8%AD%E6%96%87.txt””。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围:</b> true: 启用SDK编码。 false: 不启用SDK编码。</p> <p><b>默认取值:</b> true</p>



表 10-76 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释:</b> 获取上传进度, 详见 <a href="#">progressChanged</a> <b>默认取值:</b> 无

表 10-77 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释:</b> 传输进度数据, 详见 <a href="#">ProgressStatus</a> <b>默认取值:</b> 无

表 10-78 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 10-79 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。

常量名	原始值	说明
COLD	COLD	归档存储。

表 10-80 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释：</b> 对象数据的长度。</p> <p><b>默认取值：</b> 如果不设置，则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释：</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围：</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值：</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释：</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值：</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释：</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值：</b> 无</p>
cacheControl	String	可选	<p><b>参数解释：</b> 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><b>默认取值：</b> 无</p>

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合, 以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>
expires	String	可选	<p><b>参数解释:</b> 指对象在网页中的缓存过期时间。</p> <p><b>约束限制:</b> 日期格式为GMT的格式。</p> <p><b>默认取值:</b> 无</p>
contentTypeMD5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比, 如果不匹配, 返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>对象数据的MD5值必须经过Base64编码。</li> <li>如果不设置对象的MD5值, OBS服务端会忽略对对象数据的MD5值校验。</li> </ul> <p><b>取值范围:</b> 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: n58lG6hfM7vql4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>
storageClass	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时, 可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 10-81 DownloadFileResult

参数名称	参数类型	描述
objectMetadata	ObjectMetadata	<p><b>参数解释:</b> 对象元数据，详见<a href="#">ObjectMetadata</a>。</p>

## 代码示例

本示例展示了使用断点续传方式将examplebucket桶里的objectname对象下载到本地文件localfile中。示例代码如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.DownloadFileRequest;
import com.obs.services.model.DownloadFileResult;
public class DownloadFile001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
```

```
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk,endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    DownloadFileRequest request = new DownloadFileRequest("examplebucket", "objectname");
    // 设置下载对象的本地文件路径
    request.setDownloadFile("localfile");
    // 设置分段下载时的最大并发数
    request.setTaskNum(5);
    // 设置分段大小为10MB
    request.setPartSize(10 * 1024 * 1024);
    // 开启断点续传模式
    request.setEnableCheckpoint(true);
    // 进行断点续传下载
    DownloadFileResult result = obsClient.downloadFile(request);
    System.out.println("downloadFile successfully");
    System.out.println("Etag:" + result.getObjectMetadata().getEtag());
} catch (ObsException e) {
    System.out.println("downloadFile failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("downloadFile failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于下载对象的API说明，请参见[获取对象内容](#)。
- 更多关于下载对象的示例代码，请参见[Github示例](#)。
- 下载对象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](#)。
- 下载对象常见问题请参见[下载对象失败](#)。

## 10.10 下载对象接口实现图片处理(Java SDK)

### 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，您可以通过传入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。

更多关于图片处理的内容，参见[图片处理特性指南](#)。

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 所有的图片处理操作均不会修改存储在桶中的原图。
- 归档存储不支持图片处理。
- 深度归档存储不支持图片处理。
- 处理后的图片直接返回浏览器展示，不会保存在OBS中，也不会占用存储空间，不会产生存储费用。图片处理只收取处理的费用。

## 方法定义

obsClient.getObject([GetObjectRequest request](#))

## 请求参数说明

表 10-82 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">GetObjectRequest</a>	必选	<b>参数解释：</b> 桶下载对象时请求参数，详见 <a href="#">GetObjectRequest</a> 。

表 10-83 GetObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
imageProcess	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 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<a href="#">处理图片</a>。</p> <p><b>默认取值:</b> 如果不输入处理命令, 将返回原图。</p>
rangeStart	Long	可选	<p><b>参数解释:</b> 指定下载对象的起始位置。</p> <p><b>取值范围:</b> 非负整数。</p> <p><b>默认取值:</b> 无</p>
rangeEnd	Long	可选	<p><b>参数解释:</b> 指定下载对象的结束位置。</p> <p><b>约束限制:</b> 如果该值大于对象长度-1, 实际仍取对象长度-1。</p> <p><b>默认取值:</b> 无</p>
ifMatchTag	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
ifNoneMatchTag	String	可选	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
IfModifiedSince	Date	可选	<b>参数解释:</b> 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, 否则返回错误。 <b>默认取值:</b> 无
IfUnmodifiedSince	Date	可选	<b>参数解释:</b> 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, 否则返回错误。 <b>默认取值:</b> 无
sseCHeader	<a href="#">SseCHeader</a>	可选	<b>参数解释:</b> 服务端解密头信息, 详见 <a href="#">SseCHeader</a> 。
replaceMetadata	<a href="#">ObjectReplaceMetadata</a>	可选	<b>参数解释:</b> 指定OBS返回请求的相关信息, 详见 <a href="#">ObjectReplaceMetadata</a> 。
progressListener	<a href="#">ProgressListener</a>	可选	<b>参数解释:</b> 设置数据传输监听器, 用于获取下载进度, 详见 <a href="#">ProgressListener</a> 。

表 10-84 ObjectRepleaceMetadata

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 无</p>
contentLanguage	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>
expires	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示或在访问时弹出文件下载对话框。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
contentEncoding	String	可选	<b>参数解释:</b> 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。 <b>默认取值:</b> 无

表 10-85 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释:</b> 获取上传进度, 详见 <a href="#">progressChanged</a> <b>默认取值:</b> 无

表 10-86 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释:</b> 传输进度数据, 详见 <a href="#">ProgressStatus</a> <b>默认取值:</b> 无

表 10-87 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。

方法名称	返回值类型	说明
getTransferredBytes() ( )	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 10-88 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard, AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥，byte[]形式，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥，由原始密钥经过Base64编码后得到，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 10-89 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 10-90 ServerAlgorithm

常量值	原始值
AES256	AES256

表 10-91 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 10-92 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 10-93 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<b>参数解释:</b> 对象数据的长度。 <b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。
contentType	String	可选	<b>参数解释:</b> 对象的文件类型 ( MIME类型 )。 contentType ( 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。 <b>取值范围:</b> 常见的contentType ( MIME ) 列表参见 <a href="#">如何理解Content-Type ( MIME ) ? (Java SDK)</a> 。 <b>默认取值:</b> 如果未指定Content-Type, SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type ( 如.xml判断为application/xml文件; .html判断为text/html文件 )。
contentEncoding	String	可选	<b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。 <b>默认取值:</b> 无
contentDisposition	String	可选	<b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象, 当下载对象或者访问对象时, 以默认文件名命名的文件将直接在浏览器上显示, 或在弹出文件下载对话框时显示。 <b>默认取值:</b> 无
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>
expires	String	可选	<p><b>参数解释:</b> 指对象在网页中的缓存过期时间。</p> <p><b>约束限制:</b> 日期格式为GMT的格式。</p> <p><b>默认取值:</b> 无</p>
contentTypeMD5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>对象数据的MD5值必须经过Base64编码。</li> <li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li> </ul> <p><b>取值范围:</b> 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：n58IG6hfM7vqI4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>
storageClass	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。 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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li><li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li></ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

getObject返回结果如下

表 10-94 ObsObject

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
owner	<b>Owner</b>	<p><b>参数解释:</b> 所有者，详见<b>Owner</b>。</p>
metadata	<b>ObjectMetadata</b>	<p><b>参数解释:</b> 对象元数据，详见<b>ObjectMetadata</b>。</p>
objectContent	InputStream	<p><b>参数解释:</b> 对象数据流。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

图片处理下载。以下代码展示对examplebucket桶里的objectname.jpg图片对象进行缩放、旋转然后进行下载。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.GetObjectRequest;
import com.obs.services.model.ObsObject;
public class GetObject008 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 发起图片处理请求并下载
            // examplebucket为桶名，objectname.jpg为对象名
            GetObjectRequest request = new GetObjectRequest("examplebucket", "objectname.jpg");
            // 设置图片处理参数，对图片依次进行缩放、旋转
            request.setImageProcess("image/resize,m_fixed,w_100,h_100/rotate,90");
            ObsObject obsObject = obsClient.getObject(request);
            System.out.println("getObject successfully");
            obsObject.getObjectContent().close();
        } catch (ObsException e) {
            System.out.println("getObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("getObject failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于图片处理的更多信息，请参见[图片处理](#)。

- 图片处理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](#)。
- 图片处理常见问题请参见[图片处理常见问题](#)。

## 10.11 临时授权方式实现图片处理(Java SDK)

### 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。您可以通过临时授权方式传入图片处理参数，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。

更多关于图片处理的内容，参见[图片处理特性指南](#)。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 所有的图片处理操作均不会修改存储在桶中的原图。
- 归档存储不支持图片处理。
- 深度归档存储不支持图片处理。
- 处理后的图片直接返回浏览器展示，不会保存在OBS中，也不会占用存储空间，不会产生存储费用。图片处理只收取处理的费用。

### 方法定义

obsClient.createTemporarySignature([TemporarySignatureRequest request](#))

### 请求参数说明

表 10-95 createTemporarySignature 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">TemporarySignatureRequest</a>	必选	<b>参数解释：</b> 临时url创建请求参数列表，详见 <a href="#">TemporarySignatureRequest</a> 。

表 10-96 TemporarySignatureRequest

参数名称	参数类型	是否必选	描述
bucketName	String	可选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	可选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
specialParam	<b>SpecialParamEnum</b>	可选	<p><b>参数解释:</b> 请求中可能用到的特殊参数，详见 <b>SpecialParamEnum</b>。</p>

参数名称	参数类型	是否必选	描述
method	<a href="#">HttpMethodEnum</a>	必选	<b>参数解释:</b> HTTP方法类型, 详见 <a href="#">HttpMethodEnum</a> 。
headers	Map<String, String>	可选	<b>参数解释:</b> HTTP响应消息头列表, 由多个元组构成。元组中第一个String代表响应消息头的名称, 第二个String代表响应消息头的值。 <b>默认取值:</b> 无
queryParams	Map<String, Object>	可选	<b>参数解释:</b> 请求中携带的查询参数。Map中类型为String的key代表request中的查询参数, 类型为Object的value是查询参数取得的值。 <b>默认取值:</b> 无
expires	long	必选	<b>参数解释:</b> 表示对象的过期时间(从对象最后修改时间开始计算), 单位是天。过期之后对象会被自动删除。 <b>约束限制:</b> 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。 <b>取值范围:</b> 大于0的整型数, 单位: 天。 <b>默认取值:</b> 无
requestDate	Date	必选	<b>参数解释:</b> 发起请求的时间。 <b>默认取值:</b> 无

表 10-97 HttpMethodEnum & SpecialParamEnum

操作名	HTTP请求方法 (OBS Java SDK对应值)	特殊操作符 (OBS Java SDK对应值)	是否需要桶名	是否需要对象名
创建桶	HttpMethodEnum.PUT	N/A	是	否

操作名	HTTP请求方法 ( OBS Java SDK对 应值 )	特殊操作符 ( OBS Java SDK对应值 )	是否需 要桶名	是否需 要对象名
获取桶列表	HttpMethodEnum. GET	N/A	否	否
删除桶	HttpMethodEnum. DELETE	N/A	是	否
列举桶内对 象	HttpMethodEnum. GET	N/A	是	否
列举桶内多 版本对象	HttpMethodEnum. GET	SpecialParamEnum. VERSIONS	是	否
列举分段上 传任务	HttpMethodEnum. GET	SpecialParamEnum. UPLOADS	是	否
获取桶元数 据	HttpMethodEnum. HEAD	N/A	是	否
获取桶区域 位置	HttpMethodEnum. GET	SpecialParamEnum.L OCATION	是	否
获取桶存量 信息	HttpMethodEnum. GET	SpecialParamEnum.S TORAGEINFO	是	否
设置桶配额	HttpMethodEnum.P UT	SpecialParamEnum. QUOTA	是	否
获取桶配额	HttpMethodEnum. GET	SpecialParamEnum. QUOTA	是	否
设置桶存储 类别	HttpMethodEnum.P UT	SpecialParamEnum.S TORAGEPOLICY	是	否
获取桶存储 类别	HttpMethodEnum. GET	SpecialParamEnum.S TORAGEPOLICY	是	否
设置桶访问 权限	HttpMethodEnum.P UT	SpecialParamEnum. ACL	是	否
获取桶访问 权限	HttpMethodEnum. GET	SpecialParamEnum. ACL	是	否
开启/关闭桶 日志	HttpMethodEnum.P UT	SpecialParamEnum.L OGGING	是	否
查看桶日志	HttpMethodEnum. GET	SpecialParamEnum.L OGGING	是	否
设置桶策略	HttpMethodEnum.P UT	SpecialParamEnum.P OLICY	是	否
查看桶策略	HttpMethodEnum. GET	SpecialParamEnum.P OLICY	是	否



操作名	HTTP请求方法 ( OBS Java SDK对应值 )	特殊操作符 ( OBS Java SDK对应值 )	是否需要桶名	是否需要对象名
删除桶策略	HttpMethodEnum.DELETE	SpecialParamEnum.POLICY	是	否
设置生命周期规则	HttpMethodEnum.PUT	SpecialParamEnum.LIFECYCLE	是	否
查看生命周期规则	HttpMethodEnum.GET	SpecialParamEnum.LIFECYCLE	是	否
删除生命周期规则	HttpMethodEnum.DELETE	SpecialParamEnum.LIFECYCLE	是	否
设置托管配置	HttpMethodEnum.PUT	SpecialParamEnum.WEBSITE	是	否
查看托管配置	HttpMethodEnum.GET	SpecialParamEnum.WEBSITE	是	否
清除托管配置	HttpMethodEnum.DELETE	SpecialParamEnum.WEBSITE	是	否
设置桶多版本状态	HttpMethodEnum.PUT	SpecialParamEnum.VERSIONING	是	否
查看桶多版本状态	HttpMethodEnum.GET	SpecialParamEnum.VERSIONING	是	否
设置跨域规则	HttpMethodEnum.PUT	SpecialParamEnum.CORS	是	否
查看跨域规则	HttpMethodEnum.GET	SpecialParamEnum.CORS	是	否
删除跨域规则	HttpMethodEnum.DELETE	SpecialParamEnum.CORS	是	否
设置桶标签	HttpMethodEnum.PUT	SpecialParamEnum.TAGGING	是	否
查看桶标签	HttpMethodEnum.GET	SpecialParamEnum.TAGGING	是	否
删除桶标签	HttpMethodEnum.DELETE	SpecialParamEnum.TAGGING	是	否
上传对象	HttpMethodEnum.PUT	N/A	是	是
追加上传	HttpMethodEnum.POST	SpecialParamEnum.APPEND	是	是
下载对象	HttpMethodEnum.GET	N/A	是	是

操作名	HTTP请求方法 ( OBS Java SDK对应值 )	特殊操作符 ( OBS Java SDK对应值 )	是否需要桶名	是否需要对象名
复制对象	HttpMethodEnum.PUT	N/A	是	是
删除对象	HttpMethodEnum.DELETE	N/A	是	是
批量删除对象	HttpMethodEnum.POST	SpecialParamEnum.DELETE	是	是
获取对象属性	HttpMethodEnum.HEAD	N/A	是	是
设置对象访问权限	HttpMethodEnum.PUT	SpecialParamEnum.ACL	是	是
查看对象访问权限	HttpMethodEnum.GET	SpecialParamEnum.ACL	是	是
初始化分段上传任务	HttpMethodEnum.POST	SpecialParamEnum.UPLOADS	是	是
上传段	HttpMethodEnum.PUT	N/A	是	是
复制段	HttpMethodEnum.PUT	N/A	是	是
列举已上传的段	HttpMethodEnum.GET	N/A	是	是
合并段	HttpMethodEnum.POST	N/A	是	是
取消分段上传任务	HttpMethodEnum.DELETE	N/A	是	是
恢复归档存储对象	HttpMethodEnum.POST	SpecialParamEnum.RESTORE	是	是

## 返回结果说明

表 10-98 TemporarySignatureResponse

参数名称	参数类型	描述
signedUrl	String	<b>参数解释:</b> 带授权信息的URL。 <b>默认取值:</b> 无

参数名称	参数类型	描述
actualSignedRequest Headers	Map<String, String>	<p><b>参数解释:</b> 通过带授权信息的URL发起请求时实际应携带的头域。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

以下代码展示了通过临时授权方式实现图片处理，对examplebucket桶里的objectname.jpg对象进行缩放，旋转。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import java.util.HashMap;
import java.util.Map;
public class GetObject009 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 通过临时授权方式实现图片处理
            long expireSeconds = 3600L;
            TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.GET,
            expireSeconds);
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname.jpg");
            // 设置图片处理参数，对图片依次进行缩放、旋转
            Map<String, Object> queryParams = new HashMap<String, Object>();
            queryParams.put("x-image-process", "image/resize,m_fixed,w_100,h_100/rotate,90");
            request.setQueryParams(queryParams);
            // 生成临时授权URL
            TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
            System.out.println("getSignedUrl successfully");
            System.out.println(response.getSignedUrl());
        } catch (ObsException e) {
            System.out.println("getSignedUrl failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
        }
    }
}
```

```
// 请求失败,打印详细错误信息
System.out.println("Error Message:" + e.getErrorMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("getSignedUrl failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于图片处理的更多信息，请参见[图片处理](#)。
- 图片处理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](#)。
- 图片处理常见问题请参见[图片处理常见问题](#)。

# 11 管理对象(Java SDK)

## 11.1 设置对象元数据(Java SDK)

### 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，设置对象的元数据信息。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 一个对象可以有多个元数据，总大小不能超过8KB。
- 当前元数据名称不支持非ASCII码字符，元数据值包含非ASCII码字符时需进行Base64编码。
- 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对象，不能对其设置对象的元数据。
- 当文件夹或目录是实际的对象时，对象桶支持对其元数据的修改；如果是创建对象test/test1.txt时带的模拟目录test/，对象桶不支持模拟目录修改元数据。

## 方法定义

obsClient.setObjectMetadata([SetObjectMetadataRequest request](#))

## 请求参数说明

表 11-1 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">SetObjectMetadataRequest</a>	必选	<b>参数解释：</b> 设置对象元数据请求参数，详见 <a href="#">表 11-2</a> 。

表 11-2 SetObjectMetadataRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释：</b> 桶名。 <b>约束限制：</b> <ul style="list-style-type: none"><li>• 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。</li><li>• 桶命名规则如下：<ul style="list-style-type: none"><li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>- 禁止使用IP地址。</li><li>- 禁止以“-”或“.”开头及结尾。</li><li>- 禁止两个“.”相邻（如：“my.bucket”）。</li><li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>• 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。</li></ul> <b>默认取值：</b> 无

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
storageClass	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 指定对象的存储类别。</p> <p><b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>
removeUnset	boolean	可选	<p><b>参数解释:</b> 以前设置的元数据是否被移除。</p> <p><b>取值范围:</b> true: 设置为true后，此次设置元数据请求调用成功后，会把其他以前设置的元数据删除（标准元数据不受影响）。 false: 设置为false后，此次设置元数据请求调用成功后，保留以前设置元数据（标准元数据不受影响）。</p> <p><b>默认取值:</b> false</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, String>	可选	<p><b>参数解释:</b> 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。Map中第一个String代表以“x-obs-meta-”开头的自定义元数据名称，第二个String代表自定义元数据的值。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><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> <li>• 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。例如x-obs-meta-中文：中文 经URL编码后发送，“中文”的URL编码为： %E4%B8%AD%E6%96%87，则响应为x-obs-meta-%E4%B8%AD%E6%96%87: %E4%B8%AD%E6%96%87。</li> </ul> <p><b>默认取值:</b> 无</p>
userHeaders	HashMap<String, String>	可选	<p><b>参数解释:</b> 用户头域列表。HashMap中第一个String代表用户头域名称，第二个String代表用户头域对应的取值。通过每个SDK设置的用户头域userHeader透传给服务端，使SDK不做处理，让后续使用更灵活。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制,无法发送非ASCII码字符,SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码,发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”,则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围:</b> true: 启用SDK编码。 false: 不启用SDK编码。</p> <p><b>默认取值:</b> true</p>
replaceMetadata	<b>ObjectReplaceMetadata</b>	可选	<p><b>参数解释:</b> 指定Object被下载时的相关元数据,详见<b>ObjectReplaceMetadata</b>。</p>
websiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头,长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>

表 11-3 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 11-4 ObjectRepleaceMetadata

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 无</p>
contentType	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>
expires	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
contentDisposition	String	可选	<b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示或在访问时弹出文件下载对话框。 <b>默认取值:</b> 无
contentEncoding	String	可选	<b>参数解释:</b> 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。 <b>默认取值:</b> 无

## 返回结果说明

表 11-5 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<b>参数解释:</b> 对象数据的长度。 <b>约束限制:</b> <ul style="list-style-type: none"><li>• 单次上传对象大小范围是[0, 5GB]。</li><li>• 如果需要上传超过5GB的大文件，需要通过<a href="#">多段操作</a>来分段上传。</li></ul> <b>默认取值:</b> 如果不设置，则SDK会自动计算对象数据的长度。
contentType	String	可选	<b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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。 <b>取值范围:</b> 常见的contentType（MIME）列表参见 <a href="#">如何理解Content-Type（MIME）？(Java SDK)</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentEncoding	String	可选	<p><b>参数解释:</b> 指定对象被下载时的内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示或在访问时弹出文件下载对话框。</p> <p><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 指定对象被下载时的网页的缓存行为。</p> <p><b>默认取值:</b> 无</p>
contentType	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>
expires	String	可选	<p><b>参数解释:</b> 指对象在网页中的缓存过期时间。</p> <p><b>约束限制:</b> 日期格式为GMT的格式。</p> <p><b>默认取值:</b> 无</p>
contentMd5	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 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><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
storageClass	StorageClassEnum	可选	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 可选择的存储类别参见StorageClassEnum。</p> <p><b>默认取值:</b> 无</p>
websiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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，该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头，长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度，单位：字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为Appendable对象。</p> <p><b>取值范围:</b> true：是，可追加上传。 false：否，不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。</p> <p><b>默认取值:</b> 无</p>
statusCode	int	不需要手动设置，作为请求响应时自动设置	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	不需要手动设置，作为请求响应时自动设置	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例：设置对象元数据

通过自定义元数据和利用用户头域userHeader透传两种方式，来展示如何设置examplebucket桶下objectname对象的元数据。


```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObjectMetadata;
import com.obs.services.model.SetObjectMetadataRequest;
import java.util.HashMap;
public class SetObjectMetadata001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
```

```
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk,endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 设置对象元数据
    SetObjectMetadataRequest request = new SetObjectMetadataRequest("examplebucket",
"objectname");
    // 对象的 HTTP 标准头域
    request.setContentType("ContentType");
    request.setExpires("Expires");
    // 设置自定义元数据时，SDK会自动拼接x-obs-meta-。
    request.addUserMetadata("property1", "property-value1");
    HashMap<String, String> userHeaders = new HashMap<>();
    // SDK设置的用户头域userHeader是透传的，不做处理可让后续使用更灵活，所以如果userHeader需要在服务端以自定义元数据形式显示。在设置的用户头域userHeader前，请加上x-obs-meta-，便于服务端识别。
    userHeaders.put("property2", "property-value2");
    userHeaders.put("x-obs-meta-property3", "property-value3");
    request.setUserHeaders(userHeaders);
    ObjectMetadata metadata = obsClient.setObjectMetadata(request);
    System.out.println("setObjectMetadata successfully");
    System.out.println("UserMetadata:" + metadata.getUserMetadata("property1"));
} catch (ObsException e) {
    System.out.println("setObjectMetadata failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("setObjectMetadata failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

运行以上示例代码，设置自定义元数据property1和用户头域property2、x-obs-meta-property3后，网络抓包HTTP协议的头域如下图所示：

```
PUT /?metadata HTTP/1.1
x-obs-meta-property1: property-value1
x-obs-metadata-directive: REPLACE_NEW
property2: property-value2
x-obs-meta-property3: property-value3
```

## 代码示例：删除对象自定义元数据

如果需要删除某些自定义元数据，在调用设置对象元数据接口时，把需要保留的元数据设置一次，并设置SetObjectMetadataRequest.setRemoveUnset(true)。此时调用接口后，未设置元数据的数据会被删除，所以请慎重使用SetObjectMetadataRequest.setRemoveUnset(true)。

以下示例为删除examplebucket桶下objectname对象的自定义元数据，请参考以下示例代码：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.SetObjectMetadataRequest;
import java.util.HashMap;
public class SetObjectMetadata002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            SetObjectMetadataRequest request = new SetObjectMetadataRequest();
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname");
            request.setRemoveUnset(true);
            HashMap<String, String> userHeaders = new HashMap<>();
            // 设置request.setRemoveUnset(true)后，服务端会保存元数据property1，原有已经保存的自定义元数据会被删除。
            userHeaders.put("x-obs-meta-property1", "property-value1");
            request.setUserHeaders(userHeaders);
            obsClient.setObjectMetadata(request);
            System.out.println("setObjectMetadata successfully");
        } catch (ObsException e) {
            System.out.println("setObjectMetadata failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("setObjectMetadata failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于设置对象的元数据的API说明，请参见[修改对象元数据](#)。
- 更多关于设置对象元数据的示例代码，请参见[Github示例](#)。
- 设置对象元数据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](#)。



## 11.2 获取对象元数据(Java SDK)

### 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。

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。

该操作支持服务端加密功能。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 您也可以通过metadata.getOriginalHeaders获取所有原始响应头的信息。

### 方法定义

obsClient.getObjectMetadata([GetObjectMetadataRequest request](#))

### 请求参数说明

表 11-6 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">GetObjectMetadataRequest</a>	必选	<b>参数解释：</b> 获取对象元数据请求参数，详见 <a href="#">表11-7</a> 。

表 11-7 GetObjectMetadataRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userHeaders	HashMap<String, String>	可选	<p><b>参数解释:</b> 用户头域列表。HashMap中第一个String代表用户头域名称，第二个String代表用户头域对应的取值。通过每个SDK设置的用户头域userHeader透传给服务端，使SDK不做处理，让后续使用更灵活。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否开启OBS对请求头域的自动编码。由于HTTP编码规范限制，无法发送非ASCII码字符，SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码，发送编码后数据。如您设置的值content-disposition为“attachment; filename="中文.txt"”，则对象元数据中存储的信息为“attachment; filename="%E4%B8%AD%E6%96%87.txt"”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围:</b> true: 启用SDK编码。 false: 不启用SDK编码。</p> <p><b>默认取值:</b> true</p>
sseHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端解密头信息。详见<a href="#">SseCHeader</a>。</p>

表 11-8 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard, AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥，byte[]形式，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥，由原始密钥经过Base64编码后得到，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 11-9 ServerAlgorithm

常量值	原始值
AES256	AES256

表 11-10 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

## 返回结果说明

表 11-11 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<b>参数解释:</b> 对象数据的长度。 <b>约束限制:</b> <ul style="list-style-type: none"><li>• 单次上传对象大小范围是[0, 5GB]。</li><li>• 如果需要上传超过5GB的大文件，需要通过<a href="#">多段操作</a>来分段上传。</li></ul> <b>默认取值:</b> 如果不设置，则SDK会自动计算对象数据的长度。
contentType	String	可选	<b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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。 <b>取值范围:</b> 常见的contentType（MIME）列表参见 <a href="#">如何理解Content-Type（MIME）？（Java SDK）</a> 。 <b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。
contentEncoding	String	可选	<b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentDisposition	String	可选	<b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象, 当下载对象或者访问对象时, 以默认文件名命名的文件将直接在浏览器上显示, 或在弹出文件下载对话框时显示。 <b>默认取值:</b> 无
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无
contentType	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合, 以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
contentMd5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <p><b>取值范围:</b> 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：n58lG6hfM7vql4K0vnWpog==</p> <p><b>默认取值:</b> 无</p>
storageClass	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。 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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 11-12 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

## 代码示例

本示例展示了如何获取examplebucket桶下objectname对象的元数据。示例代码如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObjectMetadata;
public class GetObjectMetadata001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");
```

```
// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 获取对象元数据
    ObjectMetadata metadata = obsClient.getObjectMetadata("examplebucket", "objectname");
    System.out.println("getObjectMetadata successfully");
    System.out.println("ContentType:" + metadata.getContentType());
    System.out.println("ContentLength:" + metadata.getContentLength());
    // 获取用户自定义元数据
    System.out.println("UserMetadata:" + metadata.getUserMetadata("property"));
    // 获取所有原始响应头域
    System.out.println("OriginalHeaders:" + metadata.getOriginalHeaders());
} catch (ObsException e) {
    System.out.println("getObjectMetadata failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getObjectMetadata failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于获取对象元数据的API说明, 请参见[获取对象元数据](#)。
- 更多关于获取对象元数据的示例代码, 请参见[Github示例](#)。
- 获取对象元数据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](#)。

## 11.3 设置对象 ACL(Java SDK)

### 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)。

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

对象访问权限与桶访问权限类似，也支持[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](#)
2. [调用ObsClient.setObjectAcl指定预定义访问策略](#)
3. [调用ObsClient.setObjectAcl直接设置](#)

## 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以及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([SetObjectAclRequest request](#))

## 请求参数说明

表 11-13 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">SetObjectAclRequest</a>	必选	参数解释： 设置对象ACL请求参数，详见 <a href="#">表11-14</a> 。

表 11-14 SetObjectAclRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my.bucket”和“my.bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
acl	<a href="#">AccessControlList</a>	必选	<p><b>参数解释:</b> 指定对象的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>

表 11-15 ACL 预定义访问策略

常量名	说明
AccessControlList.REST_CANNED_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。
AccessControlList.REST_CANNED_PUBLIC_READ	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。 设在对象上，所有人可以获取该对象内容和元数据。
AccessControlList.REST_CANNED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControlList.REST_CANNED_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，可以获取该桶内对象的内容和元数据。 不能应用于对象。

常量名	说明
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 11-16 AccessControlList

参数名称	参数类型	是否必选	描述
owner	<b>Owner</b>	必选	<b>参数解释：</b> 桶所有者的信息，详见 <b>Owner</b> 。
delivered	boolean	可选	<b>参数解释：</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。 <b>取值范围：</b> true：是，桶ACL向桶内对象传递。 false：否，桶ACL不向桶内对象传递，仅作用于桶。 <b>默认取值：</b> false
grants	Set< <b>GrantAndPermission</b> >	可选	<b>参数解释：</b> 被授权用户相关信息，详见 <b>GrantAndPermission</b> 。

表 11-17 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释：</b> 桶所有者的账号ID，即domain_id。 <b>取值范围：</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值：</b> 无

参数名称	参数类型	是否必选	描述
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 11-18 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<p><b>参数解释:</b> 被授权用户或用户组，详见<a href="#">GranteeInterface</a>。</p>
permission	<a href="#">Permission</a>	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 11-19 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<p><b>参数解释:</b> 被授权用户的用户信息，详见<a href="#">CanonicalGrantee</a>。</p>

参数名称	参数类型	是否必选	描述
<b>GroupGrantee</b>	<b>GroupGrantee</b>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <b>GroupGrantee</b> 。 <b>默认取值:</b> 无

表 11-20 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 11-21 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。



表 11-22 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。
PERMISSION_WRITE_ACP	WRITE_ACP	修改ACL配置的权限。 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。
PERMISSION_FULL_CONTROL	FULL_CONTROL	完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。 如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。 如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。

## 返回结果说明

表 11-23 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

### 代码示例：上传对象时指定预定义访问策略

以下代码展示如何将本地文件localfile上传到examplebucket桶下的objectname对象，同时指定预定义访问策略：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
import com.obs.services.model.PutObjectRequest;
import java.io.File;
public class PutObject012 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
```

```
// 上传对象时指定预定义访问策略
PutObjectRequest request = new PutObjectRequest();
request.setBucketName("examplebucket");
request.setObjectKey("objectname");
request.setFile(new File("localfile"));
// 设置对象访问权限为私有读写
request.setAcl(AccessControlList.REST_CANNED_PRIVATE);
obsClient.putObject(request);
System.out.println("putObject successfully");
} catch (ObsException e) {
    System.out.println("putObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("putObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：为对象设置预定义访问策略

以下代码展示如何将examplebucket桶下的objectname对象通过预定义访问策略设置访问权限为私有读写：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
public class SetObjectAcl001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 为对象设置预定义访问策略
            // 设置对象访问权限为私有读写
            obsClient.setObjectAcl("examplebucket", "objectname", AccessControlList.REST_CANNED_PRIVATE);
            System.out.println("setObjectAcl successfully");
        } catch (ObsException e) {
            System.out.println("setObjectAcl failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
```

```
System.out.println("Error Code:" + e.getErrorCode());
// 请求失败,打印详细错误信息
System.out.println("Error Message:" + e.getErrorMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("setObjectAcl failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：直接设置对象访问权限

以下代码展示如何将examplebucket桶下的objectname对象直接设置对象访问权限：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
import com.obs.services.model.CanonicalGrantee;
import com.obs.services.model.GroupGrantee;
import com.obs.services.model.Owner;
import com.obs.services.model.Permission;
public class SetObjectAcl002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            AccessControlList acl = new AccessControlList();
            Owner owner = new Owner();
            owner.setId("ownerid");
            // 租户id，必填
            acl.setOwner(owner);
            // 保留Owner的完全控制权限（注：如果不设置该权限，该对象Owner自身将没有访问权限）
            acl.grantPermission(new CanonicalGrantee("ownerid"), Permission.PERMISSION_FULL_CONTROL);
            // 为指定用户设置完全控制权限
            acl.grantPermission(new CanonicalGrantee("userid"), Permission.PERMISSION_FULL_CONTROL);
            // 为所有用户设置读权限
            acl.grantPermission(GroupGrantee.ALL_USERS, Permission.PERMISSION_READ);
            obsClient.setObjectAcl("examplebucket", "objectname", acl);
            System.out.println("setObjectAcl successfully");
            System.out.println(acl);
        } catch (ObsException e) {
            System.out.println("setObjectAcl failed");
            // 请求失败,打印Http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
        }
    }
}
```

```
// 请求失败,打印详细错误信息
System.out.println("Error Message:" + e.getErrorMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("setObjectAcl failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置对象ACL的API说明, 请参见[设置对象ACL](#)。
- 更多关于设置对象ACL的示例代码, 请参见[Github示例](#)。
- 设置对象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](#)。

## 11.4 获取对象 ACL(Java SDK)

### 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。可以通过本接口获取指定桶中对象的访问权限。

### 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以及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([GetObjectAclRequest request](#))

### 请求参数说明

表 11-24 请求参数列表

参数名称	参数类型	是否必选	描述
request	GetObjectAclRequest	必选	<b>参数解释:</b> 获取对象ACL请求参数, 详见 <a href="#">GetObjectAclRequest</a> 。

表 11-25 GetObjectAclRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-.bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 11-26 AccessControlList

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
owner	<a href="#">Owner</a>	<b>参数解释:</b> 对象的所有者。详见 <a href="#">Owner</a> 。
grants	Set< <a href="#">GrantAndPermission</a> >	<b>参数解释:</b> 被授权用户相关信息，详见 <a href="#">GrantAndPermission</a> 。
delivered	boolean	<b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。 <b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。 <b>默认取值:</b> false

表 11-27 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组, 详见 <a href="#">GranteeInterface</a> 。
permission	<a href="#">Permission</a>	必选	<b>参数解释:</b> 用户或用户组被授予的权限。 <b>取值范围:</b> 详见 <a href="#">Permission</a> 。 <b>默认取值:</b> 无
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false

表 11-28 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无



表 11-29 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 11-30 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 11-31 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 11-32 Permission

常量名	原始值	说明
PERMISSION_READ	READ	<p>读权限。</p> <p>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。</p> <p>如果有对象的读权限，则可以获取该对象内容和元数据。</p>
PERMISSION_WRITE	WRITE	<p>写权限。</p> <p>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。</p> <p>此权限在对象上不适用。</p>
PERMISSION_READ_ACP	READ_ACP	<p>读取ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有读对应桶或对象ACP的权限。</p>
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>

常量名	原始值	说明
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

## 代码示例

以下代码展示如何获取examplebucket桶下的objectname对象的访问权限。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
public class GetObjectAcl001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 获取对象访问权限
            AccessControlList acl = obsClient.getObjectAcl("examplebucket", "objectname");
            System.out.println("getObjectAcl successfully");
            System.out.println(acl);
        } catch (ObsException e) {
            System.out.println("getObjectAcl failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("getObjectAcl failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

```
}  
}  
}
```

## 相关链接

- 关于获取对象ACL的API说明，请参见[获取对象ACL](#)。
- 更多关于设置对象ACL的示例代码，请参见[Github示例](#)。
- 设置对象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](#)。

## 11.5 列举对象(Java SDK)

### 功能说明

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

- 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 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，可以采用[分页列举](#)的方式返回桶中所有多版本对象。列举时，ListVersionsResult.isTruncated为true表明本次没有返回全部对象，可以通过ListVersionsResult.getNextKeyMarker和ListVersionsResult.getNextVersionIdMarker获取下次列举的起始位置。

### 方法定义

obsClient.listObjects([ListObjectsRequest request](#))

### 请求参数说明

表 11-33 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">ListObjectsRequest</a>	必选	<b>参数解释：</b> 列举对象请求参数，详见 <a href="#">ListObjectsRequest</a> 。

表 11-34 ListObjectsRequest

参数	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
prefix	String	可选	<p><b>参数解释:</b> 列举桶内对象列表时，指定一个前缀，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><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数	参数类型	是否必选	描述
marker	String	可选	<p><b>参数解释:</b> 列举桶内对象列表时，指定一个标识符，作为列举时的起始位置，从该标识符以后按对象名的字典顺序返回对象列表。</p> <p>例如，您拥有以下对象：test/a、test/b、test/c、test/d。如果您将test/b指定为标识符，将返回test/c、test/d两个对象。</p> <p><b>约束限制:</b> 仅用于非多版本列举。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
maxKeys	int	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 1~1000，当超出范围时，按照默认的1000进行处理。</p> <p><b>默认取值:</b> 1000</p>

参数	参数类型	是否必选	描述
delimiter	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
encodingType	String	可选	<p><b>参数解释:</b> 对响应中的部分元素进行指定类型的编码。如果 delimiter、marker、prefix、nextMarker 和 objectKey 包含xml 1.0标准不支持的控制字符，可通过设置encodingType 对响应中的 delimiter、marker、prefix（包括commonPrefixes 中的prefix）、nextMarker 和 objectKey 进行编码。</p> <p><b>取值范围:</b> 可选值为url。</p> <p><b>默认取值:</b> 无，不设置则不编码。</p>

## 返回结果说明

表 11-35 ObjectListing

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
prefix	String	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
objectSummaries	List<ObsObject>	<p><b>参数解释:</b> 桶内对象列表，详见<a href="#">表11-36</a>。</p>



参数名称	参数类型	描述
commonPrefixes	List<String>	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
truncated	boolean	<p><b>参数解释:</b> 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><b>取值范围:</b> true: 表示没有返回全部结果。 false: 表示已返回了全部结果。</p> <p><b>默认取值:</b> 无</p>
marker	String	<p><b>参数解释:</b> 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。例如, 您拥有以下对象: test/a、test/b、test/c、test/d。如果您将test/b指定为标识符, 将返回test/c、test/d两个对象。</p> <p><b>约束限制:</b> 仅用于非多版本列举。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
maxKeys	int	<p><b>参数解释:</b> 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> <p><b>取值范围:</b> 1~1000, 当超出范围时, 按照默认的1000进行处理。</p> <p><b>默认取值:</b> 1000</p>

参数名称	参数类型	描述
delimiter	String	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
nextMarker	String	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
location	String	<p><b>参数解释:</b> 桶所在的区域。</p> <p><b>约束限制:</b> 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><b>取值范围:</b> 当前有效的OBS区域和终端节点的更多信息，请参考<a href="#">地区和终端节点</a>。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><b>默认取值:</b> 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>
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

表 11-36 ObsObject

参数名称	参数类型	描述
objectKey	String	<p><b>参数解释:</b> 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.cn- north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串</p> <p><b>默认取值:</b> 无</p>
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectContent	InputStream	<p><b>参数解释:</b> 获取对象的数据流。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
owner	<b>Owner</b>	<p><b>参数解释:</b> 对象的所有者，包含对象拥有者DomainId和对象拥有者名称，详见<b>Owner</b>。</p> <p><b>默认取值:</b> 无。</p>
metadata	<b>ObjectMetadata</b>	<p><b>参数解释:</b> 对象的元数据信息。</p> <p><b>取值范围:</b> 对象的元数据信息，详见<b>ObjectMetadata</b>。</p> <p><b>默认取值:</b> 无</p>

表 11-37 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID，即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a></p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a></p> <p><b>默认取值:</b> 无</p>

表 11-38 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 11-39 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 单次上传对象大小范围是[0, 5GB]。</li> <li>• 如果需要上传超过5GB的大文件，需要通过<a href="#">多段操作</a>来分段上传。</li> </ul> <p><b>默认取值:</b> 如果不设置，则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentType	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>
expires	String	可选	<p><b>参数解释:</b> 指对象在网页中的缓存过期时间。</p> <p><b>约束限制:</b> 日期格式为GMT的格式。</p> <p><b>默认取值:</b> 无</p>
contentMD5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>对象数据的MD5值必须经过Base64编码。</li> <li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li> </ul> <p><b>取值范围:</b> 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><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
storageClass	StorageClassEnum	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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，该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>必须以“/”、“http://”或“https://”开头，长度不超过2KB。</li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度，单位：字节。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称, Object代表自定义元数据的值。对象的自定义元数据可以通过 <a href="#">ObsClient.getObjectMetadata</a> 获取, 参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据, 总大小不能超过8KB。</li> <li>• 使用 <a href="#">ObsClient.getObject</a> 下载对象时, 对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

## 代码示例：简单列举对象

以下代码展示如何简单列举examplebucket桶下的对象, 最多返回1000个对象:

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObjectListing;
import com.obs.services.model.ObsObject;
public class ListObjects001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK, 也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken, 也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint, 此处以华北-北京四为例, 其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint, 也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
```

```
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 简单列举
    ObjectListing result = obsClient.listObjects("examplebucket");
    for (ObsObject obsObject : result.getObjects()) {
        System.out.println("listObjects successfully");
        System.out.println("ObjectKey:" + obsObject.getObjectKey());
        System.out.println("Owner:" + obsObject.getOwner());
    }
} catch (ObsException e) {
    System.out.println("listObjects failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("listObjects failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：指定数目列举对象

以下代码展示如何指定数目列举examplebucket桶下的对象，列举最大个数为100：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListObjectsRequest;
import com.obs.services.model.ObjectListing;
import com.obs.services.model.ObsObject;
public class ListObjects002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 指定数目列举
            ListObjectsRequest request = new ListObjectsRequest("examplebucket");
            // 只列举100个对象
            request.setMaxKeys(100);
            ObjectListing result = obsClient.listObjects(request);
            for (ObsObject obsObject : result.getObjects()) {
```

```
        System.out.println("listObjects successfully");
        System.out.println("ObjectKey:" + obsObject.getObjectKey());
        System.out.println("Owner:" + obsObject.getOwner());
    }
} catch (ObsException e) {
    System.out.println("listObjects failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("listObjects failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：指定前缀列举对象

以下代码展示如何指定前缀prefix列举examplebucket桶下的对象，列举最大个数为100：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListObjectsRequest;
import com.obs.services.model.ObjectListing;
import com.obs.services.model.ObsObject;
public class ListObjects003 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 指定前缀列举
            ListObjectsRequest request = new ListObjectsRequest("examplebucket");
            // 设置列举带有prefix前缀的100个对象
            request.setMaxKeys(100);
            request.setPrefix("prefix");
            ObjectListing result = obsClient.listObjects(request);
            for (ObsObject obsObject : result.getObjects()) {
                System.out.println("listObjects successfully");
                System.out.println("ObjectKey:" + obsObject.getObjectKey());
                System.out.println("Owner:" + obsObject.getOwner());
            }
        } catch (ObsException e) {
```

```
System.out.println("listObjects failed");
// 请求失败,打印http状态码
System.out.println("HTTP Code:" + e.getResponseCode());
// 请求失败,打印服务端错误码
System.out.println("Error Code:" + e.getErrorCode());
// 请求失败,打印详细错误信息
System.out.println("Error Message:" + e.getErrorMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("listObjects failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：指定起始位置列举对象

以下代码展示如何指定起始位置列举examplebucket桶下的对象，列举最大个数为100：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListObjectsRequest;
import com.obs.services.model.ObjectListing;
import com.obs.services.model.ObsObject;
public class ListObjects004 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 指定起始位置列举
            ListObjectsRequest request = new ListObjectsRequest("examplebucket");
            // 设置列举对象名字典序在"test"之后的100个对象
            request.setMaxKeys(100);
            request.setMarker("test");
            ObjectListing result = obsClient.listObjects(request);
            for (ObsObject obsObject : result.getObjects()) {
                System.out.println("listObjects successfully");
                System.out.println("ObjectKey:" + obsObject.getObjectKey());
                System.out.println("Owner:" + obsObject.getOwner());
            }
        } catch (ObsException e) {
            System.out.println("listObjects failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
        }
    }
}
```

```
// 请求失败,打印详细错误信息
System.out.println("Error Message:" + e.getErrorMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("listObjects failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：分页列举全部对象

以下代码展示分页列举examplebucket桶下的全部对象，设置每页最多列举100个对象：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListObjectsRequest;
import com.obs.services.model.ObjectListing;
import com.obs.services.model.ObsObject;
public class ListObjects005 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 分页列举全部对象
            ListObjectsRequest request = new ListObjectsRequest("examplebucket");
            // 设置每页100个对象
            request.setMaxKeys(100);
            ObjectListing result;
            do {
                result = obsClient.listObjects(request);
                for (ObsObject obsObject : result.getObjects()) {
                    System.out.println("listObjects successfully");
                    System.out.println("ObjectKey:" + obsObject.getObjectKey());
                    System.out.println("Owner:" + obsObject.getOwner());
                }
                request.setMarker(result.getNextMarker());
            } while (result.isTruncated());
        } catch (ObsException e) {
            System.out.println("listObjects failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
        }
    }
}
```

```
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("listObjects failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：列举文件夹中的对象

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：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListObjectsRequest;
import com.obs.services.model.ObjectListing;
import com.obs.services.model.ObsObject;
public class ListObjects006 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 列举文件夹中的所有对象
            ListObjectsRequest request = new ListObjectsRequest("examplebucket");
            // 设置文件夹对象名"dir/"为前缀
            request.setPrefix("dir/");
            request.setMaxKeys(100);
            ObjectListing result;
            do {
                result = obsClient.listObjects(request);
                for (ObsObject obsObject : result.getObjects()) {
                    System.out.println("listObjects successfully");
                    System.out.println("ObjectKey:" + obsObject.getObjectKey());
                    System.out.println("Owner:" + obsObject.getOwner());
                }
                request.setMarker(result.getNextMarker());
            } while (result.isTruncated());
        } catch (ObsException e) {
            System.out.println("listObjects failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
        }
    }
}
```

```
        System.out.println("Error Message:" + e.getMessage());
        // 请求失败,打印请求id
        System.out.println("Request ID:" + e.getErrorRequestId());
        System.out.println("Host ID:" + e.getErrorHostId());
        e.printStackTrace();
    } catch (Exception e) {
        System.out.println("listObjects failed");
        // 其他异常信息打印
        e.printStackTrace();
    }
}
```

## 代码示例：按文件夹分组，列举桶内所有对象

以下代码展示如何按文件夹分组，列举examplebucket桶下的所有对象：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListObjectsRequest;
import com.obs.services.model.ObjectListing;
import com.obs.services.model.ObsObject;
public class ListObjects007 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 按文件夹分组列举所有对象
            ListObjectsRequest request = new ListObjectsRequest("examplebucket");
            request.setMaxKeys(1000);
            // 设置文件夹分隔符"/"
            request.setDelimiter("/");
            ObjectListing result = obsClient.listObjects(request);
            System.out.println("Objects in the root directory:");
            for (ObsObject obsObject : result.getObjects()) {
                System.out.println("listObjects successfully");
                System.out.println("ObjectKey:" + obsObject.getObjectKey());
                System.out.println("Owner:" + obsObject.getOwner());
            }
            listObjectsByPrefix(obsClient, request, result);
        } catch (ObsException e) {
            System.out.println("listObjects failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
        }
    }
}
```

```
e.printStackTrace();
} catch (Exception e) {
    System.out.println("listObjects failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：递归列出子文件夹中对象

递归列出子文件夹中对象的函数 *listObjectsByPrefix* 的示例代码如下：

```
// 递归列出子文件夹中对象
static void listObjectsByPrefix(ObsClient obsClient, ListObjectsRequest request, ObjectListing result)
    throws ObsException {
    for (String prefix : result.getCommonPrefixes()) {
        System.out.println("Objects in folder [" + prefix + "]:");
        request.setPrefix(prefix);
        result = obsClient.listObjects(request);
        for (ObsObject obsObject : result.getObjects()) {
            System.out.println("ObjectKey:" + obsObject.getObjectKey());
            System.out.println("Owner:" + obsObject.getOwner());
        }
        listObjectsByPrefix(obsClient, request, result);
    }
}
```

### 📖 说明

- 以上代码示例不包含文件夹中对象数超过1000个的情况。
- 由于是需要列举出文件夹中的对象和子文件夹，且文件夹对象总是以“/”结尾，因此 delimiter 总是为“/”。
- 每次递归的返回结果中 ObjectListing.getObjects 包含的是文件夹中的对象；ObjectListing.getCommonPrefixes 包含的是文件夹的子文件夹。

## 相关链接

- 关于列举对象的其他说明，请参见[授权码列举对象](#)。
- 更多关于列举对象的示例代码，请参见[Github 示例](#)。
- 获取桶列表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](#)。

## 11.6 删除对象(Java SDK)

### 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对象。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。



## 接口约束

- 您必须是桶所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以及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([DeleteObjectRequest request](#))

## 请求参数说明

表 11-40 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">DeleteObjectRequest</a>	必选	<b>参数解释：</b> 删除对象请求参数，详见 <a href="#">DeleteObjectRequest</a> 。

表 11-41 DeleteObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 11-42 DeleteObjectResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
versionId	String	<b>参数解释:</b> 对象的版本号。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
deleteMarker	boolean	<b>参数解释:</b> 标识删除的对象是否是删除标记。 <b>取值范围:</b> <ul style="list-style-type: none"><li>• true: 是删除标记。</li><li>• false: 不是删除标记。</li></ul> <b>默认取值:</b> false

参数名称	参数类型	描述
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

删除单个对象。以下代码示例展示删除examplebucket桶下objectname对象。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class DeleteObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 删除单个对象
            obsClient.deleteObject("examplebucket", "objectname");
            System.out.println("deleteObject successfully");
        } catch (ObsException e) {
            System.out.println("deleteObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
```

```
System.out.println("deleteObject failed");
// 其他异常信息打印
e.printStackTrace();
}
}
```

## 相关链接

- 关于删除对象的API说明，请参见[删除对象](#)。
- 更多关于删除对象的示例代码，请参见[Github示例](#)。
- 删除对象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](#)。

## 11.7 批量删除对象(Java SDK)

### 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以及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([DeleteObjectsRequest deleteRequest](#))

### 请求参数说明

表 11-43 请求参数列表

参数名称	参数类型	是否必选	描述
deleteRequest	<a href="#">DeleteObjectsRequest</a>	必选	<b>参数解释：</b> 批量删除对象请求参数，详见 <a href="#">DeleteObjectsRequest</a> 。

表 11-44 DeleteObjectsRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
keyAndVersions	List<KeyAndVersion>	必选	<p><b>参数解释:</b> 待删除的对象列表，详情参考<a href="#">KeyAndVersion</a>。</p>
quiet	boolean	可选	<p><b>参数解释:</b> 批量删除对象的响应方式。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>false: 表示详细模式，返回删除成功和删除失败的所有结果。</li> <li>true: 表示简单模式，只返回删除过程中出错的结果。</li> </ul> <p><b>默认取值:</b> false</p>

参数名称	参数类型	是否必选	描述
encodingType	String	可选	<p><b>参数解释:</b> 用于指定对响应中的对象名objectKey进行指定类型的编码。如果objectKey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objectKey进行编码。</p> <p><b>取值范围:</b> 可选值为url。</p> <p><b>默认取值:</b> 无, 不设置则不编码。</p>

表 11-45 KeyAndVersion

参数名称	参数类型	是否必选	描述
key	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中, 对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号, 用于删除指定版本号的对象。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无, 如果不设置则默认删除最新版本的对象。</p>

## 返回结果说明

表 11-46 DeleteObjectsResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
deletedObjectResults	List< <a href="#">DeleteObjectResult</a> >	<p><b>参数解释:</b> 删除对象后的响应结果，详见<a href="#">DeleteObjectResult</a>。</p>
errorResults	List< <a href="#">ErrorResult</a> >	<p><b>参数解释:</b> 删除失败的对象列表，详见<a href="#">ErrorResult</a>。</p>

表 11-47 DeleteObjectResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
versionId	String	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
deleteMarker	boolean	<p><b>参数解释:</b> 标识删除的对象是否是删除标记。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• true: 是删除标记。</li> <li>• false: 不是删除标记。</li> </ul> <p><b>默认取值:</b> false</p>
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

表 11-48 ErrorResult

参数名称	参数类型	描述
versionId	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
errorCode	String	<p><b>参数解释:</b> 删除失败的错误码。可参考<a href="#">OBS错误码</a>。</p>
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
message	String	<p><b>参数解释:</b> 删除失败的错误原因。可参考<a href="#">OBS错误码</a>。</p>

## 代码示例

以下代码展示了如何进行批量删除examplebucket桶下的所有对象。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.DeleteObjectsRequest;
import com.obs.services.model.DeleteObjectsResult;
import com.obs.services.model.ListVersionsRequest;
import com.obs.services.model.ListVersionsResult;
import com.obs.services.model.VersionOrDeleteMarker;
import java.awt.*;
public class DeleteObjects001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
    }
}
```

```
// 您可以登录访问管理控制台获取访问密钥AK/SK
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
// 【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/SecurityToken，也可以使用其他外部引入方式传入。
// String securityToken = System.getenv("SECURITY_TOKEN");
// endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk,endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 批量删除对象
    ListVersionsRequest request = new ListVersionsRequest("examplebucket");
    // 每次批量删除100个对象
    request.setMaxKeys(100);
    ListVersionsResult result;
    do {
        result = obsClient.listVersions(request);
        DeleteObjectsRequest deleteRequest = new DeleteObjectsRequest("examplebucket");
        // deleteRequest.setQuiet(true); // 注意此demo默认是详细模式，如果要使用简单模式，请添加本行
        for (VersionOrDeleteMarker v : result.getVersions()) {
            // 获取对象和版本号
            deleteRequest.addKeyAndVersion(v.getKey(), v.getVersionId());
        }
        if(deleteRequest.getKeyAndVersions().length > 0){
            DeleteObjectsResult deleteResult = obsClient.deleteObjects(deleteRequest);
            // 获取删除成功的对象
            System.out.println("DeletedObjectResults:"+ deleteResult.getDeletedObjectResults());
            // 获取删除失败的对象
            System.out.println("ErrorResults:"+ deleteResult.getErrorResults());
        }else{
            System.out.println("No object to delete");
        }
        request.setKeyMarker(result.getNextKeyMarker());
        // 如果没有开启多版本对象，就不需要设置VersionIdMarker
        request.setVersionIdMarker(result.getNextVersionIdMarker());
    } while (result.isTruncated());
    System.out.println("deleteObjects successfully");
} catch (ObsException e) {
    System.out.println("deleteObjects failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("deleteObjects failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

代码

## 相关链接

- 关于批量删除对象的API说明，请参见[批量删除对象](#)。
- 更多关于删除对象的示例代码，请参见[Github示例](#)。
- 删除对象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](#)。

## 11.8 复制对象(Java SDK)

### 功能说明

用户可以根据需要将存储在OBS上的对象复制到其他路径下。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。

为指定桶中的对象创建一个副本。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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](#)。
- 用户有待复制的源对象的读权限。
- OBS支持的region以及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拷贝的原桶和对象信息，不能携带消息实体。
- 支持同区域跨桶复制，不支持跨区域复制。
- 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[分段复制](#)功能复制对象。
- 如果待复制的源对象是归档存储类别，则必须先恢复源对象才能进行复制。

### 方法定义

obsClient.copyObject([CopyObjectRequest request](#))

## 请求参数说明

表 11-49 请求参数列表

参数名称	参数类型	是否必选	描述
request	<b>CopyObjectRequest</b>	必选	<b>参数解释：</b> 复制对象请求参数，详见 <b>CopyObjectRequest</b> 。

表 11-50 CopyObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释：</b> 目标桶名。 <b>约束限制：</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <b>默认取值：</b> 无

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 目标对象名。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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
sseKmsHeader	<a href="#">SseKmsHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息，用于加密目标对象。详见<a href="#">SseKmsHeader</a>。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息，用于加密目标对象。详见<a href="#">SseCHeader</a>。</p> <p><b>默认取值:</b> 无</p>
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 指定对象的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>

参数名称	参数类型	是否必选	描述
successRedirectLocation	String	可选	<p><b>参数解释:</b> 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。</p> <ul style="list-style-type: none"><li>如果此参数值有效且操作成功，响应码为303，返回值中的Location头域由此参数以及桶名、对象名、对象的ETag组成。</li><li>如果此参数值无效，则OBS忽略此参数的作用，返回值中的Location头域为对象地址，响应码根据操作成功或失败正常返回。</li></ul> <p><b>默认取值:</b> 无</p>
sourceBucketName	String	必选	<p><b>参数解释:</b> 源桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sourceObjectKey	String	必选	<p><b>参数解释:</b> 源对象名。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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
newObjectMetadata	<b>ObjectMetadata</b>	可选	<p><b>参数解释:</b> 目标对象的自定义元数据，详见 <b>ObjectMetadata</b>。</p> <p><b>约束限制:</b> replaceMetadata需与 newObjectMetadata配合使用。</p> <p><b>默认取值:</b> 无</p>
replaceMetadata	boolean	可选	<p><b>参数解释:</b> 复制对象时，是否重写源对象的元数据。</p> <p><b>约束限制:</b> replaceMetadata需与 newObjectMetadata配合使用。</p> <p><b>取值范围:</b> true: 重写源对象的元数据。 false: 不重写源对象的元数据。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
ifModifiedSince	java.util.Date	可选	<p><b>参数解释:</b> 如果源对象的修改时间早于该参数值指定的时间, 则进行复制, 否则返回错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 如果包含ifUnmodifiedSince并且不符合, 或者包含ifMatchTag并且不符合, 或者包含ifModifiedSince并且不符合, 或者包含ifNoneMatchTag并且不符合, 则复制失败, 抛出异常中HTTP状态码为: 412 precondition failed。</li><li>• ifModifiedSince和ifNoneMatchTag可以一起使用; ifUnmodifiedSince和ifMatchTag可以一起使用。</li></ul> <p><b>默认取值:</b> 无</p>
ifUnmodifiedSince	java.util.Date	可选	<p><b>参数解释:</b> 如果源对象的修改时间晚于该参数值指定的时间, 则进行复制, 否则返回错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 如果包含ifUnmodifiedSince并且不符合, 或者包含ifMatchTag并且不符合, 或者包含ifModifiedSince并且不符合, 或者包含ifNoneMatchTag并且不符合, 则复制失败, 抛出异常中HTTP状态码为: 412 precondition failed。</li><li>• ifModifiedSince和ifNoneMatchTag可以一起使用; ifUnmodifiedSince和ifMatchTag可以一起使用。</li></ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
ifMatchTag	String	可选	<p><b>参数解释:</b> 指定一个预设的Etag值, 如果源对象的Etag值与该参数值相同, 则返回对象内容, 否则返回错误。其中, 源对象的Etag值是指源对象数据的MD5校验值。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 如果包含ifUnmodifiedSince并且不符合, 或者包含ifMatchTag并且不符合, 或者包含ifModifiedSince并且不符合, 或者包含ifNoneMatchTag并且不符合, 则复制失败, 抛出异常中HTTP状态码为: 412 precondition failed。</li><li>• ifModifiedSince和ifNoneMatchTag可以一起使用; ifUnmodifiedSince和ifMatchTag可以一起使用。</li></ul> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
ifNoneMatchTag	String	可选	<p><b>参数解释:</b> 指定一个预设的Etag值, 如果源对象的Etag值与该参数值不相同, 则返回对象内容, 否则返回错误。其中, 源对象的Etag值是指源对象数据的MD5校验值。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 如果包含ifUnmodifiedSince并且不符合, 或者包含ifMatchTag并且不符合, 或者包含ifModifiedSince并且不符合, 或者包含ifNoneMatchTag并且不符合, 则复制失败, 抛出异常中HTTP状态码为: 412 precondition failed。</li><li>• ifModifiedSince和ifNoneMatchTag可以一起使用; ifUnmodifiedSince和ifMatchTag可以一起使用。</li></ul> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
versionId	String	可选	<p><b>参数解释:</b> 源对象版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
sseCHeaderSource	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端解密头信息，用于解密源对象。详见<a href="#">SseCHeader</a>。</p> <p><b>默认取值:</b> 无</p>

表 11-51 SseKmsHeader

参数名称	参数类型	是否必选	描述
encryption	<a href="#">ServerEncryption</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p><b>取值范围:</b> 可选值：kms，即选择SSE-KMS方式加密对象，详见<a href="#">ServerEncryption</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持kms。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
kmsKeyId	String	可选	<p><b>参数解释:</b> SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p><b>取值范围:</b> 有效值支持两种格式:</p> <ol style="list-style-type: none"> <li>regionID:domainID(账号ID):key/key_id</li> <li>key_id</li> </ol> <p>其中:</p> <ul style="list-style-type: none"> <li>regionID是使用密钥所属region的ID,可在<a href="#">地区和终端节点</a>页面获取;</li> <li>domainID是使用密钥所属账号的账号ID,获取方法参见<a href="#">如何获取账号ID和用户ID?</a>;</li> <li>key_id是从数据加密服务创建的密钥ID,获取方法请参见<a href="#">查看密钥</a>。</li> </ul> <p><b>默认取值:</b></p> <ul style="list-style-type: none"> <li>如果用户没有提供该头域,那么默认的主密钥将会被使用。</li> <li>如果默认主密钥不存在,将默认创建并使用。</li> </ul>

表 11-52 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式,对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值: AES256, 即选择SSE-C方式,使用高级加密标准(Advanced Encryption Standard, AES)加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<b>参数解释:</b> 加密算法。 <b>约束限制:</b> 只支持AES256。 <b>取值范围:</b> 详见 <a href="#">SSEAlgorithmEnum</a> 。 <b>默认取值:</b> 无
sseCKey	byte[]	必选	<b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。 <b>默认取值:</b> 无
sseCKeyBase64	String	可选	<b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过Base64编码后得到, 该密钥用于加密对象。 <b>默认取值:</b> 无

表 11-53 ServerEncryption

常量值	原始值
OBS_KMS	kms

表 11-54 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 11-55 ServerAlgorithm

常量值	原始值
AES256	AES256

表 11-56 ACL 预定义访问策略

常量名	说明
AccessControlList.REST_CANNED_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。
AccessControlList.REST_CANNED_PUBLIC_READ	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。 设在对象上，所有人可以获取该对象内容和元数据。
AccessControlList.REST_CANNED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControlList.REST_CANNED_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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControlList.REST_CANNED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

表 11-57 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	Owner	可选	参数解释： 桶所有者的信息，详见Owner。

参数名称	参数类型	是否必选	参数类型
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。 <b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。 <b>默认取值:</b> false
grants	Set< <a href="#">GrantAndPermission</a> >	可选	<b>参数解释:</b> 被授权用户相关信息，详见 <a href="#">GrantAndPermission</a> 。

表 11-58 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID，即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 11-59 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组，详见 <a href="#">GranteeInterface</a> 。

参数名称	参数类型	是否必选	描述
permission	Permission	必选	<p><b>参数解释：</b> 用户或用户组被授予的权限。</p> <p><b>取值范围：</b> 详见Permission。</p> <p><b>默认取值：</b> 无</p>
delivered	boolean	可选	<p><b>参数解释：</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围：</b> true：是，桶ACL向桶内对象传递。 false：否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值：</b> false</p>

表 11-60 Permission

常量名	原始值	说明
PERMISSION_READ	READ	<p>读权限。</p> <p>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。</p> <p>如果有对象的读权限，则可以获取该对象内容和元数据。</p>
PERMISSION_WRITE	WRITE	<p>写权限。</p> <p>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。</p> <p>此权限在对象上不适用。</p>
PERMISSION_READ_ACP	READ_ACP	<p>读取ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有读对应桶或对象ACP的权限。</p>



常量名	原始值	说明
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

表 11-61 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<p><b>参数解释：</b> 被授权用户的用户信息，详见<a href="#">CanonicalGrantee</a>。</p>
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<p><b>参数解释：</b> 被授权用户组的用户组信息。</p> <p><b>取值范围：</b> 详见<a href="#">GroupGrantee</a>。</p> <p><b>默认取值：</b> 无</p>

表 11-62 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<p><b>参数解释:</b> 被授权用户的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
displayName	String	可选	<p><b>参数描述:</b> 被授权用户的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 11-63 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 11-64 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 单次上传对象大小范围是[0, 5GB]。</li> <li>• 如果需要上传超过5GB的大文件, 需要通过<a href="#">多段操作</a>来分段上传。</li> </ul> <p><b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。</p>

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？（Java SDK）</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentTypeLanguage	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无
contentMd5	String	可选	<b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。 <b>约束限制:</b> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <b>取值范围:</b> 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：n58IG6hfM7vqI4K0vnWpog== <b>默认取值:</b> 无
storageClasses	<a href="#">StorageClassEnum</a>	可选	<b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。 <b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 11-65 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

## 返回结果说明

表 11-66 CopyObjectResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
lastModified	java.util.Date	<p><b>参数解释:</b> 目标对象的最近一次修改时间。</p> <p><b>默认取值:</b> 无</p>
etag	String	<p><b>参数解释:</b> 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。上传的对象或拷贝操作创建的对象，都有唯一的ETag。</p> <p><b>约束限制:</b> 当对象是服务端加密的对象时，ETag值不是对象的MD5值。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	<p><b>参数解释:</b> 目标对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
copySourceVersionId	String	<p><b>参数解释:</b> 源对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
storageClass	<b>StorageClassEnum</b>	<b>参数解释:</b> 目标对象的存储类别。 <b>取值范围:</b> 详见 <b>StorageClassEnum</b> 。 <b>默认取值:</b> 无

## 代码示例：简单复制对象

本示例展示了如何将sourcebucketname桶下的sourceobjectname对象简单复制到destbucketname桶下的destobjectname对象中。示例代码如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.CopyObjectResult;
public class CopyObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 简单复制
            CopyObjectResult result =
                obsClient.copyObject("sourcebucketname", "sourceobjectname", "destbucketname",
"destobjectname");
            System.out.println("copyObject successfully");
            System.out.println("StatusCode:" + result.getStatusCode());
            System.out.println("Etag:" + result.getEtag());
        } catch (ObsException e) {
            System.out.println("copyObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("copyObject failed");
        }
    }
}
```



```
        // 其他异常信息打印
        e.printStackTrace();
    }
}
}
```

## 代码示例：复制对象时重写对象属性

以下代码展示了如何在复制对象时重写对象属性，将sourcebucketname桶下的sourceobjectname对象复制到destbucketname桶下的destobjectname对象中。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.CopyObjectRequest;
import com.obs.services.model.CopyObjectResult;
import com.obs.services.model.ObjectMetadata;
import com.obs.services.model.StorageClassEnum;
public class CopyObject002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 复制对象时重写对象属性
            CopyObjectRequest request = new CopyObjectRequest("sourcebucketname", "sourceobjectname",
"destbucketname", "destobjectname");
            // 设置进行对象属性重写
            request.setReplaceMetadata(true);
            HashMap<String, String> userHeaders = new HashMap<>();
            userHeaders.put("Content-Type", "image/jpeg");
            request.setUserHeaders(userHeaders);
            CopyObjectResult result = obsClient.copyObject(request);
            System.out.println("copyObject successfully");
            System.out.println("Etag:" + result.getEtag());
        } catch (ObsException e) {
            System.out.println("copyObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("copyObject failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

```
}  
}
```

## 代码示例：限定条件复制

限定条件复制。将sourcebucketname桶下的sourceobjectname对象复制到destbucketname桶下的destobjectname对象中。

```
import com.obs.services.ObsClient;  
import com.obs.services.exception.ObsException;  
import com.obs.services.model.CopyObjectRequest;  
import com.obs.services.model.CopyObjectResult;  
import java.text.SimpleDateFormat;  
public class CopyObject003 {  
    public static void main(String[] args) {  
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。  
        // 您可以登录访问管理控制台获取访问密钥AK/SK  
        String ak = System.getenv("ACCESS_KEY_ID");  
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");  
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。  
        // String securityToken = System.getenv("SECURITY_TOKEN");  
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。  
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";  
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。  
        //String endPoint = System.getenv("ENDPOINT");  
  
        // 创建ObsClient实例  
        // 使用永久AK/SK初始化客户端  
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);  
        // 使用临时AK/SK和SecurityToken初始化客户端  
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);  
  
        try {  
            // 限定条件复制  
            CopyObjectRequest request =  
                new CopyObjectRequest("sourcebucketname", "sourceobjectname", "destbucketname",  
"destobjectname");  
            request.setIfModifiedSince(new SimpleDateFormat("yyyy-MM-dd").parse("2016-01-01"));  
            request.setIfNoneMatchTag("none-match-etag");  
            CopyObjectResult result = obsClient.copyObject(request);  
            System.out.println("copyObject successfully");  
            System.out.println("Etag:" + result.getEtag());  
        } catch (ObsException e) {  
            System.out.println("copyObject failed");  
            // 请求失败,打印http状态码  
            System.out.println("HTTP Code:" + e.getResponseCode());  
            // 请求失败,打印服务端错误码  
            System.out.println("Error Code:" + e.getErrorCode());  
            // 请求失败,打印详细错误信息  
            System.out.println("Error Message:" + e.getErrorMessage());  
            // 请求失败,打印请求id  
            System.out.println("Request ID:" + e.getErrorRequestId());  
            System.out.println("Host ID:" + e.getErrorHostId());  
            e.printStackTrace();  
        } catch (Exception e) {  
            System.out.println("copyObject failed");  
            // 其他异常信息打印  
            e.printStackTrace();  
        }  
    }  
}
```

## 代码示例：复制对象时重写对象访问权限

复制对象时重写对象访问权限，将sourcebucketname桶下的sourceobjectname对象复制到destbucketname桶下的destobjectname对象中。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
import com.obs.services.model.CopyObjectRequest;
import com.obs.services.model.CopyObjectResult;
public class CopyObject004 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 复制对象时重写对象访问权限
            CopyObjectRequest request =
                new CopyObjectRequest("sourcebucketname", "sourceobjectname", "destbucketname",
"destobjectname");
            // 复制时重写对象访问权限为私有读写
            request.setAcl(AccessControlList.REST_CANNED_PRIVATE);
            CopyObjectResult result = obsClient.copyObject(request);
            System.out.println("copyObject successfully");
            System.out.println("Etag:" + result.getEtag());
        } catch (ObsException e) {
            System.out.println("copyObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("copyObject failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于复制对象的API说明，请参见[复制对象](#)。
- 更多关于复制对象的示例代码，请参见[Github示例](#)。

- 复制对象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](#)。
- 复制对象常见问题请参见[我可以在桶间进行文件复制吗？](#)。

## 11.9 分段复制(Java SDK)

### 功能说明

分段复制是分段上传的一种特殊情况，即分段上传任务中的段可以通过复制OBS指定桶中现有对象（或对象的一部分）来实现。

您可以通过ObsClient.copyPart来复制段。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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](#)。
- 用户有待复制的源对象的读权限。
- OBS支持的region以及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拷贝的原桶和对象信息，不能携带消息实体。
- 支持同区域跨桶复制，不支持跨区域复制。
- 如果待复制的源对象是归档存储类别，则必须先恢复源对象才能进行复制。

### 方法定义

obsClient.copyPart([CopyPartRequest request](#))

### 分段复制请求参数

表 11-67 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">CopyPartRequest</a>	必选	<b>参数解释：</b> 分段复制对象请求参数，详见 <a href="#">CopyPartRequest</a> 。

表 11-68 CopyPartRequest

参数名称	参数类型	是否必选	描述
sourceBucketName	String	必选	<p><b>参数解释:</b> 源桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
sourceObjectKey	String	必选	<p><b>参数解释:</b> 源对象名。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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
destinationBucketName	String	必选	<p><b>参数解释:</b> 目标桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my.bucket”和“my.bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
destinationObjectKey	String	必选	<p><b>参数解释:</b> 目标对象名。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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
byteRangeStart	Long	可选	<p><b>参数解释:</b> 指定复制源对象的起始位置。</p> <p><b>取值范围:</b> 非负整数，单位：字节。</p> <p><b>默认取值:</b> 0</p>

参数名称	参数类型	是否必选	描述
byteRangeEnd	Long	可选	<p><b>参数解释:</b> 指定复制源对象的结束位置。</p> <p><b>约束限制:</b> 取值必须大于RangeStart，如果该值大于对象长度-1，实际仍取对象长度-1，单位为字节。</p> <p><b>取值范围:</b> 非负整数，单位：字节。</p> <p><b>默认取值:</b> 无</p>
sseCHeaderSource	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息，用于解密源对象。详情参见<a href="#">SseCHeader</a>。</p>
sseCHeaderDestination	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息，用于加密目标对象。详情参见<a href="#">SseCHeader</a>。</p> <p><b>约束限制:</b> 如果客户端的对象上传时，使用了客户提供的加密密钥进行服务端加密，当下载对象时，同样也必须在消息中提供密钥。</p>
versionId	String	可选	<p><b>参数解释:</b> 源对象的版本号。例如： G001117FCE89978B0000401205D5DC9A。如果源对象存在多个版本，可以指定该参数。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
partNumber	int	必选	<p><b>参数解释:</b> 段号。</p> <p><b>取值范围:</b> 取值范围是[1,10000]的非负整数。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
uploadId	String	必选	<p><b>参数解释:</b> 分段上传任务的ID, 例如: 000001648453845DBB78F2340DD460D8。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

表 11-69 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式, 对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值: AES256, 即选择SSE-C方式, 使用高级加密标准 (Advanced Encryption Standard, AES) 加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
sseCKeyBase64	String	可选	<b>参数解释:</b> SSE-C方式下的密钥，由原始密钥经过Base64编码后得到，该密钥用于加密对象。 <b>默认取值:</b> 无

表 11-70 ServerAlgorithm

常量值	原始值
AES256	AES256

表 11-71 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

## 分段复制返回结果说明

表 11-72 CopyPartResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
partNumber	int	<p><b>数解释:</b> 段号。</p> <p><b>取值范围:</b> 取值范围是[1,10000]的非负整数。</p> <p><b>默认取值:</b> 无</p>
etag	String	<p><b>参数解释:</b> 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。上传的对象或拷贝操作创建的对象，都有唯一的etag。</p> <p><b>约束限制:</b> 当对象是服务端加密的对象时，etag值不是对象的MD5值。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
lastModified	java.util.Date	<p><b>参数解释:</b> 目标段的最近一次修改时间。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于复制段，将sourcebucketname源桶下的sourceobjectname对象复制到destbucketname目标桶下的destobjectname对象中。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.CompleteMultipartUploadRequest;
import com.obs.services.model.CopyPartRequest;
import com.obs.services.model.CopyPartResult;
```

```
import com.obs.services.model.InitiateMultipartUploadRequest;
import com.obs.services.model.InitiateMultipartUploadResult;
import com.obs.services.model.ObjectMetadata;
import com.obs.services.model.PartEtag;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.TimeUnit;
public class CopyPart001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 分段复制
            final String destBucketName = "destbucketname";
            final String destObjectKey = "destobjectname";
            final String sourceBucketName = "sourcebucketname";
            final String sourceObjectKey = "sourceobjectname";
            // 初始化线程池
            ExecutorService executorService = Executors.newFixedThreadPool(20);
            // 初始化分段上传任务
            InitiateMultipartUploadRequest request = new InitiateMultipartUploadRequest(destBucketName,
            destObjectKey);
            InitiateMultipartUploadResult result = obsClient.initiateMultipartUpload(request);
            final String uploadId = result.getUploadId();
            System.out.println("uploadId:" + uploadId + "\n");
            // 获取大对象信息
            ObjectMetadata metadata = obsClient.getObjectMetadata(sourceBucketName, sourceObjectKey);
            // 每段复制100MB
            long partSize = 100 * 1024 * 1024L;
            long objectSize = metadata.getContentLength();
            // 计算需要复制的段数
            long partCount = objectSize % partSize == 0 ? objectSize / partSize : objectSize / partSize + 1;
            final List<PartEtag> partEtags = Collections.synchronizedList(new ArrayList<>());
            // 执行并发复制段
            for (int i = 0; i < partCount; i++) {
                // 复制段起始位置
                final long rangeStart = i * partSize;
                // 复制段结束位置
                final long rangeEnd = (i + 1 == partCount) ? objectSize - 1 : rangeStart + partSize - 1;
                // 分段号
                final int partNumber = i + 1;
                executorService.execute(
                    new Runnable() {
                        @Override
                        public void run() {
                            CopyPartRequest request = new CopyPartRequest();
                            request.setUploadId(uploadId);
                            request.setSourceBucketName(sourceBucketName);
```

```
        request.setSourceObjectKey(sourceObjectKey);
        request.setDestinationBucketName(destBucketName);
        request.setDestinationObjectKey(destObjectKey);
        request.setByteRangeStart(rangeStart);
        request.setByteRangeEnd(rangeEnd);
        request.setPartNumber(partNumber);
        CopyPartResult result;
        try {
            result = obsClient.copyPart(request);
            System.out.println("Part#" + partNumber + " done\n");
            partEtags.add(new PartEtag(result.getEtag(), result.getPartNumber()));
        } catch (ObsException e) {
            e.printStackTrace();
        }
    }
});
// 等待复制完成
executorService.shutdown();
while (!executorService.isTerminated()) {
    try {
        executorService.awaitTermination(5, TimeUnit.SECONDS);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
// 合并段
CompleteMultipartUploadRequest completeMultipartUploadRequest =
    new CompleteMultipartUploadRequest(destBucketName, destObjectKey, uploadId, partEtags);
obsClient.completeMultipartUpload(completeMultipartUploadRequest);
System.out.println("copyObject successfully");
} catch (ObsException e) {
    System.out.println("copyObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("copyObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于分段上传-初始化分段上传任务的API说明, 请参见[初始化上传段任务](#)。
- 关于分段上传-上传段的API说明, 请参见[上传段](#)。
- 关于分段上传-合并段的API说明, 请参见[合并段](#)。
- 关于分段上传-复制段的API说明, 请参见[拷贝段](#)。
- 更多关于分段上传的示例代码, 请参见[Github示例](#)。
- 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](#)。

## 11.10 判断对象是否存在(Java SDK)

### 功能说明

判断对象是否存在，返回的结果中HTTP状态码为200表明对象存在，否则返回404表明对象或桶不存在。

### 接口约束

- 您必须是桶拥有者或拥有列举桶内对象的权限，才能判断判断对象是否存在。建议使用IAM或桶策略进行授权，如果使用IAM则需授予obs:bucket: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](#)、[配置对象策略](#)。
- OBS支持的region以及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

doesObjectExist(final [GetObjectMetadataRequest](#))

### 请求参数说明

表 11-73 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">GetObjectMetadataRequest</a>	必选	参数解释： 设置对象属性请求参数，详见 <a href="#">GetObjectMetadataRequest</a> 。

表 11-74 GetObjectMetadataRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userHeaders	HashMap<String, String>	可选	<p><b>参数解释:</b> 用户头域列表。HashMap中第一个String代表用户头域名称，第二个String代表用户头域名称对应的取值。通过每个SDK设置的用户头域userHeader透传给服务端，使SDK不做处理，让后续使用更灵活。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 设置对象元数据接口允许指定是否编解码http头域。</p> <p><b>取值范围:</b> true: 对象元数据接口允许进行编解码http头域。 false: 对象元数据接口允许不进行编解码http头域。</p> <p><b>默认取值:</b> 默认为true，默认会进行编解码。</p>
sseHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端解密头信息。详见<a href="#">SseCHeader</a>。</p>

表 11-75 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值: AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard, AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseAlgorithm	<b>SSEAlgorithmEnum</b>	可选	<b>参数解释:</b> 加密算法。 <b>约束限制:</b> 只支持AES256。 <b>取值范围:</b> 详见 <b>SSEAlgorithmEnum</b> 。 <b>默认取值:</b> 无
sseCKey	byte[]	必选	<b>参数解释:</b> SSE-C方式下加密使用的原始密钥, byte[]形式, 该密钥用于加密对象。 <b>默认取值:</b> 无
sseCKeyBase64	String	可选	<b>参数解释:</b> SSE-C方式下的密钥, 由原始密钥经过Base64编码后得到, 该密钥用于加密对象。 <b>默认取值:</b> 无

表 11-76 ServerAlgorithm

常量值	原始值
AES256	AES256

表 11-77 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256



## 返回结果说明

表 11-78 返回结果如下

方法名称	返回值类型	说明
doesObjectExist(final <a href="#">GetObjectMetadataRequest</a> request)	boolean	用于判断桶中相关对象是否存在。

## 代码示例

本示例用于判断examplebucket桶里面objectname对象是否存在。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class DoesObjectExist001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以通过登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 判断指定的对象是否存在
            System.out.println(obsClient.doesObjectExist("examplebucket", "objectname") ? "exists!" : "does not exist!");
            System.out.println("doesObjectExist successfully");
        } catch (ObsException e) {
            System.out.println("doesObjectExist failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("doesObjectExist failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 判断对象是否存在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](#)。

# 12 临时授权访问(Java SDK)

## 12.1 通过临时 URL 访问 OBS(Java SDK)

### 功能说明

OBS客户端支持通过访问密钥、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, 您需要指定URL的有效期限来限制访客用户的访问时长。

如果您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(例如使用生成PUT请求的URL上传对象), 将该URL提供给其他用户。

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。
3. 不支持通过CDN加速后的域名生成临时访问URL。

#### 须知

开发过程中, 您有任何问题可以在github上[提交issue](#), 或者在[华为云对象存储服务论坛](#)中发帖求助。

通过该方式可支持的操作以及相关信息见下表:

表 12-1 HttpMethodEnum & SpecialParamEnum

操作名	HTTP请求方法 ( OBS Java SDK对 应值 )	特殊操作符 ( OBS Java SDK对应值 )	是否需 要桶名	是否需 要对象名
创建桶	HttpMethodEnum.P UT	N/A	是	否
获取桶列表	HttpMethodEnum. GET	N/A	否	否
删除桶	HttpMethodEnum. DELETE	N/A	是	否
列举桶内对 象	HttpMethodEnum. GET	N/A	是	否
列举桶内多 版本对象	HttpMethodEnum. GET	SpecialParamEnum. VERSIONS	是	否
列举分段上 传任务	HttpMethodEnum. GET	SpecialParamEnum. UPLOADS	是	否
获取桶元数 据	HttpMethodEnum. HEAD	N/A	是	否
获取桶区域 位置	HttpMethodEnum. GET	SpecialParamEnum.L OCATION	是	否
获取桶存量 信息	HttpMethodEnum. GET	SpecialParamEnum.S TORAGEINFO	是	否
设置桶配额	HttpMethodEnum.P UT	SpecialParamEnum. QUOTA	是	否
获取桶配额	HttpMethodEnum. GET	SpecialParamEnum. QUOTA	是	否
设置桶存储 类别	HttpMethodEnum.P UT	SpecialParamEnum.S TORAGEPOLICY	是	否
获取桶存储 类别	HttpMethodEnum. GET	SpecialParamEnum.S TORAGEPOLICY	是	否
设置桶访问 权限	HttpMethodEnum.P UT	SpecialParamEnum. ACL	是	否
获取桶访问 权限	HttpMethodEnum. GET	SpecialParamEnum. ACL	是	否
开启/关闭桶 日志	HttpMethodEnum.P UT	SpecialParamEnum.L OGGING	是	否
查看桶日志	HttpMethodEnum. GET	SpecialParamEnum.L OGGING	是	否

操作名	HTTP请求方法 ( OBS Java SDK对应值 )	特殊操作符 ( OBS Java SDK对应值 )	是否需要桶名	是否需要对象名
设置桶策略	HttpMethodEnum.PUT	SpecialParamEnum.POLICY	是	否
查看桶策略	HttpMethodEnum.GET	SpecialParamEnum.POLICY	是	否
删除桶策略	HttpMethodEnum.DELETE	SpecialParamEnum.POLICY	是	否
设置生命周期规则	HttpMethodEnum.PUT	SpecialParamEnum.LIFECYCLE	是	否
查看生命周期规则	HttpMethodEnum.GET	SpecialParamEnum.LIFECYCLE	是	否
删除生命周期规则	HttpMethodEnum.DELETE	SpecialParamEnum.LIFECYCLE	是	否
设置托管配置	HttpMethodEnum.PUT	SpecialParamEnum.WEBSITE	是	否
查看托管配置	HttpMethodEnum.GET	SpecialParamEnum.WEBSITE	是	否
清除托管配置	HttpMethodEnum.DELETE	SpecialParamEnum.WEBSITE	是	否
设置桶多版本状态	HttpMethodEnum.PUT	SpecialParamEnum.VERSIONING	是	否
查看桶多版本状态	HttpMethodEnum.GET	SpecialParamEnum.VERSIONING	是	否
设置跨域规则	HttpMethodEnum.PUT	SpecialParamEnum.CORS	是	否
查看跨域规则	HttpMethodEnum.GET	SpecialParamEnum.CORS	是	否
删除跨域规则	HttpMethodEnum.DELETE	SpecialParamEnum.CORS	是	否
设置桶标签	HttpMethodEnum.PUT	SpecialParamEnum.TAGGING	是	否
查看桶标签	HttpMethodEnum.GET	SpecialParamEnum.TAGGING	是	否
删除桶标签	HttpMethodEnum.DELETE	SpecialParamEnum.TAGGING	是	否
上传对象	HttpMethodEnum.PUT	N/A	是	是

操作名	HTTP请求方法 ( OBS Java SDK对应值 )	特殊操作符 ( OBS Java SDK对应值 )	是否需要桶名	是否需要对象名
追加上传	HttpMethodEnum.POST	SpecialParamEnum.APPEND	是	是
下载对象	HttpMethodEnum.GET	N/A	是	是
复制对象	HttpMethodEnum.PUT	N/A	是	是
删除对象	HttpMethodEnum.DELETE	N/A	是	是
批量删除对象	HttpMethodEnum.POST	SpecialParamEnum.DELETE	是	是
获取对象属性	HttpMethodEnum.HEAD	N/A	是	是
设置对象访问权限	HttpMethodEnum.PUT	SpecialParamEnum.ACL	是	是
查看对象访问权限	HttpMethodEnum.GET	SpecialParamEnum.ACL	是	是
初始化分段上传任务	HttpMethodEnum.POST	SpecialParamEnum.UPLOADS	是	是
上传段	HttpMethodEnum.PUT	N/A	是	是
复制段	HttpMethodEnum.PUT	N/A	是	是
列举已上传的段	HttpMethodEnum.GET	N/A	是	是
合并段	HttpMethodEnum.POST	N/A	是	是
取消分段上传任务	HttpMethodEnum.DELETE	N/A	是	是
恢复归档存储对象	HttpMethodEnum.POST	SpecialParamEnum.RESTORE	是	是

通过OBS Java SDK生成临时URL访问OBS的步骤如下：

**步骤1** 通过ObsClient.createTemporarySignature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

## 接口约束

- OBS支持的region以及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.createTemporarySignature([TemporarySignatureRequest request](#))

## 请求参数说明

表 12-2 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">TemporarySignatureRequest</a>	必选	<b>参数解释：</b> 临时url创建请求，详见 <a href="#">TemporarySignatureRequest</a> 。

表 12-3 TemporarySignatureRequest

参数名称	参数类型	是否必选	描述
bucketName	String	可选	<b>参数解释：</b> 桶名。 <b>约束限制：</b> <ul style="list-style-type: none"><li>• 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。</li><li>• 桶命名规则如下：<ul style="list-style-type: none"><li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>- 禁止使用IP地址。</li><li>- 禁止以“-”或“.”开头及结尾。</li><li>- 禁止两个“.”相邻（如：“my..bucket”）。</li><li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>• 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。</li></ul> <b>默认取值：</b> 无

参数名称	参数类型	是否必选	描述
objectKey	String	可选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
specialParam	<a href="#">SpecialParamEnum</a>	可选	<p><b>参数解释:</b> 请求中可能用到的特殊参数，代表要操作的子资源。</p> <p><b>取值范围:</b> 详见<a href="#">SpecialParamEnum</a>。</p> <p><b>默认取值:</b> 无</p>
method	<a href="#">HttpMethodEnum</a>	必选	<p><b>参数解释:</b> HTTP方法类型。</p> <p><b>取值范围:</b> 详见<a href="#">HttpMethodEnum</a>。</p> <p><b>默认取值:</b> 无</p>
headers	Map<String, String>	可选	<p><b>参数解释:</b> 请求中携带的头域。请求头域列表Map中第一个String代表请求头域的名称，第二个String代表对应请求头域名称的取值。</p> <p><b>默认取值:</b> 无</p>
queryParams	Map<String, Object>	可选	<p><b>参数解释:</b> 请求中携带的查询参数。参数列表Map中的String为查询参数的名称，Object为对应查询参数名称的取值。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
expires	long	必选	<b>参数解释:</b> 带授权信息的URL的过期时间。 <b>取值范围:</b> 大于等于0的整型数，单位：秒。 <b>默认取值:</b> 300秒
requestDate	Date	必选	<b>参数解释:</b> 发起请求的时间。 <b>默认取值:</b> 无

## 返回结果说明

表 12-4 TemporarySignatureResponse

参数名称	参数类型	描述
signedUrl	String	<b>参数解释:</b> 带授权信息的URL。 <b>默认取值:</b> 无
actualSignedRequest Headers	Map<String, String>	<b>参数解释:</b> 通过带授权信息的URL发起请求时实际应携带的头域。头域列表Map中第一个String为请求头域的名称，第二个String为对应请求头域名称的取值。 <b>默认取值:</b> 无

## 代码示例：创建桶

临时授权访问，利用HttpMethodEnum.PUT请求创建examplebucket桶，URL有效期设置为3600秒。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import okhttp3.Call;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
```

```
import okhttp3.Response;
import java.util.Map;
public class CreateBucket001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // URL有效期，3600秒
            long expireSeconds = 3600L;
            TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.PUT,
            expireSeconds);
            request.setBucketName("examplebucket");
            TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
            System.out.println("Creating bucket using temporary signature url:");
            System.out.println("\t" + response.getSignedUrl());
            Request.Builder builder = new Request.Builder();
            for (Map.Entry<String, String> entry : response.getActualSignedRequestHeaders().entrySet()) {
                builder.header(entry.getKey(), entry.getValue());
            }
            // 使用PUT请求创建桶
            String location = "your bucket location";
            Request httpRequest =
                builder.url(response.getSignedUrl())
                    .put(
                        RequestBody.create(
                            null,
                            "<CreateBucketConfiguration><Location>"
                                + location
                                + "</Location></CreateBucketConfiguration>".getBytes())
                    ).build();
            OkHttpClient httpClient =
                new OkHttpClient.Builder()
                    .followRedirects(false)
                    .retryOnConnectionFailure(false)
                    .cache(null)
                    .build();
            Call c = httpClient.newCall(httpRequest);
            Response res = c.execute();
            System.out.println("\tStatus:" + res.code());
            if (res.body() != null) {
                System.out.println("\tContent:" + res.body().string() + "\n");
            }
            res.close();
            System.out.println("CreateBucket successfully");
        } catch (ObsException e) {
            System.out.println("CreateBucket failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
        }
    }
}
```

```
        System.out.println("Error Message:" + e.getMessage());
        // 请求失败,打印请求id
        System.out.println("Request ID:" + e.getErrorRequestId());
        System.out.println("Host ID:" + e.getErrorHostId());
        e.printStackTrace();
    } catch (Exception e) {
        System.out.println("CreateBucket failed");
        // 其他异常信息打印
        e.printStackTrace();
    }
}
```

## 代码示例：上传对象

临时授权访问，利用HttpMethodEnum.PUT请求上传对象到examplebucket桶下的objectname对象中，URL有效期设置为3600秒。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import okhttp3.Call;
import okhttp3.MediaType;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;
import java.util.HashMap;
import java.util.Map;
public class PutObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // URL有效期，3600秒
            long expireSeconds = 3600L;
            Map<String, String> headers = new HashMap<>();
            String contentType = "text/plain";
            headers.put("Content-Type", contentType);
            TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.PUT,
            expireSeconds);
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname");
            request.setHeaders(headers);
            TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
            System.out.println("Creating object using temporary signature url:");
            System.out.println("\t" + response.getSignedUrl());
            Request.Builder builder = new Request.Builder();
            for (Map.Entry<String, String> entry : response.getActualSignedRequestHeaders().entrySet()) {
```

```
        builder.header(entry.getKey(), entry.getValue());
    }
    // 使用PUT请求上传对象
    Request httpRequest =
        builder.url(response.getSignedUrl())
            .put(RequestBody.create(MediaType.parse(contentType), "Hello
OBS".getBytes("UTF-8")))
            .build();
    OkHttpClient httpClient =
        new OkHttpClient.Builder()
            .followRedirects(false)
            .retryOnConnectionFailure(false)
            .cache(null)
            .build();
    Call c = httpClient.newCall(httpRequest);
    Response res = c.execute();
    System.out.println("Status:" + res.code());
    if (res.body() != null) {
        System.out.println("Content:" + res.body().string() + "\n");
    }
    res.close();
    System.out.println("PutObject successfully");
} catch (ObsException e) {
    System.out.println("PutObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("PutObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：下载对象

临时授权访问，利用HttpMethodEnum.GET请求下载examplebucket桶下的objectname对象，URL有效期设置为3600秒。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import okhttp3.Call;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
import java.io.InputStream;
import java.util.Map;
public class GetObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
```

```
// endpoint填写桶所在的endpoint, 此处以华北-北京四为例, 其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint, 也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // URL有效期, 3600秒
    long expireSeconds = 3600L;
    TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.GET,
    expireSeconds);
    request.setBucketName("examplebucket");
    request.setObjectKey("objectname");
    TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
    System.out.println("Getting object using temporary signature url:");
    System.out.println("SignedUrl:" + response.getSignedUrl());
    Request.Builder builder = new Request.Builder();
    for (Map.Entry<String, String> entry : response.getActualSignedRequestHeaders().entrySet()) {
        builder.header(entry.getKey(), entry.getValue());
    }
    // 使用GET请求下载对象
    Request httpRequest = builder.url(response.getSignedUrl()).get().build();
    OkHttpClient httpClient =
        new OkHttpClient.Builder()
            .followRedirects(false)
            .retryOnConnectionFailure(false)
            .cache(null)
            .build();
    Call c = httpClient.newCall(httpRequest);
    Response res = c.execute();
    System.out.println("Status:" + res.code());
    InputStream objectContent = null;
    if (res.body() != null) {
        objectContent = res.body().byteStream();
        System.out.println("Content:" + res.body().string() + "\n");
    }
    if(objectContent != null) {
        // objectContent 就是获取的要下载的文件的文件流
        // 在这里可以读取 objectContent, 长时间不读取这个流会被服务端断开连接
    }
    res.close();
    System.out.println("getObject successfully");
} catch (ObsException e) {
    System.out.println("getObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：列举对象

临时授权访问，利用HttpMethodEnum.GET请求列举examplebucket桶下的对象，URL有效期设置为3600秒。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import okhttp3.Call;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
import java.util.Map;
public class ListObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // URL有效期，3600秒
            long expireSeconds = 3600L;
            TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.GET,
            expireSeconds);
            request.setBucketName("examplebucket");
            TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
            System.out.println("Getting object list using temporary signature url:");
            System.out.println("\t" + response.getSignedUrl());
            Request.Builder builder = new Request.Builder();
            for (Map.Entry<String, String> entry : response.getActualSignedRequestHeaders().entrySet()) {
                builder.header(entry.getKey(), entry.getValue());
            }
            // 使用GET请求获取对象列表
            Request httpRequest = builder.url(response.getSignedUrl()).get().build();
            OkHttpClient httpClient =
                new OkHttpClient.Builder()
                    .followRedirects(false)
                    .retryOnConnectionFailure(false)
                    .cache(null)
                    .build();
            Call c = httpClient.newCall(httpRequest);
            Response res = c.execute();
            System.out.println("Status:" + res.code());
            if (res.body() != null) {
                System.out.println("Content:" + res.body().string() + "\n");
            }
            res.close();
            System.out.println("ListObject successfully");
        } catch (ObsException e) {
            System.out.println("ListObject failed");
            // 请求失败,打印http状态码
        }
    }
}
```

```
System.out.println("HTTP Code:" + e.getResponseCode());
// 请求失败,打印服务端错误码
System.out.println("Error Code:" + e.getErrorCode());
// 请求失败,打印详细错误信息
System.out.println("Error Message:" + e.getErrorMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
System.out.println("ListObject failed");
// 其他异常信息打印
e.printStackTrace();
}
}
```

## 代码示例：删除对象

临时授权访问，利用HttpMethodEnum.DELETE请求删除examplebucket桶下的objectname对象，URL有效期设置3600秒。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import okhttp3.Call;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
import java.util.Map;
public class DeleteObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // URL有效期，3600秒
            long expireSeconds = 3600L;
            TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.DELETE,
            expireSeconds);
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname");
            TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
            System.out.println("Deleting object using temporary signature url:");
            System.out.println("\t" + response.getSignedUrl());
            Request.Builder builder = new Request.Builder();
            for (Map.Entry<String, String> entry : response.getActualSignedRequestHeaders().entrySet()) {
                builder.header(entry.getKey(), entry.getValue());
            }
            // 使用DELETE删除对象
```

```
Request httpRequest = builder.url(response.getSignedUrl()).delete().build();
OkHttpClient httpClient =
    new OkHttpClient.Builder()
        .followRedirects(false)
        .retryOnConnectionFailure(false)
        .cache(null)
        .build();
Call c = httpClient.newCall(httpRequest);
Response res = c.execute();
System.out.println("\tStatus:" + res.code());
if (res.body() != null) {
    System.out.println("\tContent:" + res.body().string() + "\n");
}
res.close();
System.out.println("deleteObjects successfully");
} catch (ObsException e) {
    System.out.println("deleteObjects failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("deleteObjects failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：初始化分段上传任务

临时授权访问，利用HttpMethodEnum.POST请求实现分段上传初始化，URL有效期设置3600秒。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.SpecialParamEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import okhttp3.Call;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;
import java.util.Map;
public class InitiateMultiPartUpload001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
```



```
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // URL有效期, 3600秒
    long expireSeconds = 3600L;
    TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.POST,
    expireSeconds);
    request.setBucketName("examplebucket");
    request.setObjectKey("objectname");
    request.setSpecialParam(SpecialParamEnum.UPLOADS);
    TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
    System.out.println("initiate multipart upload using temporary signature url:");
    System.out.println("\t" + response.getSignedUrl());
    Request.Builder builder = new Request.Builder();
    for (Map.Entry<String, String> entry : response.getActualSignedRequestHeaders().entrySet()) {
        builder.header(entry.getKey(), entry.getValue());
    }
    // 使用POST请求初始化分段上传任务
    Request httpRequest = builder.url(response.getSignedUrl()).post(RequestBody.create(null,
    "")).build();
    OkHttpClient httpClient =
        new OkHttpClient.Builder()
            .followRedirects(false)
            .retryOnConnectionFailure(false)
            .cache(null)
            .build();
    Call c = httpClient.newCall(httpRequest);
    Response res = c.execute();
    System.out.println("Status:" + res.code());
    if (res.body() != null) {
        System.out.println("Content:" + res.body().string() + "\n");
    }
    res.close();
    System.out.println("InitiateMultiPartUpload successfully");
} catch (ObsException e) {
    System.out.println("InitiateMultiPartUpload failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("InitiateMultiPartUpload failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：上传段

临时授权访问，利用HttpMethodEnum.PUT请求实现上传段，URL有效期设置3600秒。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import okhttp3.Call;
import okhttp3.OkHttpClient;
```

```
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;
import java.util.HashMap;
import java.util.Map;
public class UploadPart001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // URL有效期，3600秒
            long expireSeconds = 3600L;
            TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.PUT,
            expireSeconds);
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname");
            Map<String, Object> queryParams = new HashMap<String, Object>();
            // 设置partNumber请求参数，例如：queryParams.put("partNumber", "1");
            queryParams.put("partNumber", "partNumber");
            queryParams.put("uploadId", "your uploadId");
            request.setQueryParams(queryParams);
            TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
            System.out.println("upload part using temporary signature url:");
            System.out.println("SignedUrl:" + response.getSignedUrl());
            Request.Builder builder = new Request.Builder();
            for (Map.Entry<String, String> entry : response.getActualSignedRequestHeaders().entrySet()) {
                builder.header(entry.getKey(), entry.getValue());
            }
            // 使用PUT请求上传段
            Request httpRequest =
                builder.url(response.getSignedUrl())
                    .put(RequestBody.create(null, new byte[6 * 1024 * 1024]))
                    .build();
            OkHttpClient httpClient =
                new OkHttpClient.Builder()
                    .followRedirects(false)
                    .retryOnConnectionFailure(false)
                    .cache(null)
                    .build();
            Call c = httpClient.newCall(httpRequest);
            Response res = c.execute();
            System.out.println("Status:" + res.code());
            if (res.body() != null) {
                System.out.println("Content:" + res.body().string() + "\n");
            }
            res.close();
            System.out.println("UploadPart successfully");
        } catch (ObsException e) {
            System.out.println("UploadPart failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
        }
    }
}
```

```
// 请求失败,打印服务端错误码
System.out.println("Error Code:" + e.getErrorCode());
// 请求失败,打印详细错误信息
System.out.println("Error Message:" + e.getErrorMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("UploadPart failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：列举已上传段

临时授权访问，利用HttpMethodEnum.GET请求实现列举已上传段，URL有效期设置3600秒。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import okhttp3.Call;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
import java.util.HashMap;
import java.util.Map;
public class ListUploadedParts001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // URL有效期，3600秒
            long expireSeconds = 3600L;
            TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.GET,
            expireSeconds);
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname");
            Map<String, Object> queryParams = new HashMap<String, Object>();
            queryParams.put("uploadId", "your uploadId");
            request.setQueryParams(queryParams);
            TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
            System.out.println("list parts using temporary signature url:");
            System.out.println("\t" + response.getSignedUrl());
            Request.Builder builder = new Request.Builder();
            for (Map.Entry<String, String> entry : response.getActualSignedRequestHeaders().entrySet()) {
```

```
        builder.header(entry.getKey(), entry.getValue());
    }
    // 使用GET请求列举已上传段
    Request httpRequest = builder.url(response.getSignedUrl()).get().build();
    OkHttpClient httpClient =
        new OkHttpClient.Builder()
            .followRedirects(false)
            .retryOnConnectionFailure(false)
            .cache(null)
            .build();
    Call c = httpClient.newCall(httpRequest);
    Response res = c.execute();
    System.out.println("Status:" + res.code());
    if (res.body() != null) {
        System.out.println("Content:" + res.body().string() + "\n");
    }
    res.close();
    System.out.println("ListParts successfully");
} catch (ObsException e) {
    System.out.println("ListParts failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("ListParts failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：合并段

临时授权访问，利用HttpMethodEnum.POST请求实现合并段，URL有效期设置3600秒。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import okhttp3.Call;
import okhttp3.MediaType;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;
import java.util.HashMap;
import java.util.Map;
public class CompleteMultiPartUpload001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
```

```
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk,endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // URL有效期，3600秒
    long expireSeconds = 3600L;
    TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.POST,
expireSeconds);
    request.setBucketName("examplebucket");
    request.setObjectKey("objectname");
    Map<String, String> headers = new HashMap<>();
    String contentType = "application/xml";
    headers.put("Content-Type", contentType);
    request.setHeaders(headers);
    Map<String, Object> queryParams = new HashMap<>();
    queryParams.put("uploadId", "your uploadId");
    request.setQueryParams(queryParams);
    TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
    System.out.println("complete multipart upload using temporary signature url:");
    System.out.println("\t" + response.getSignedUrl());
    Request.Builder builder = new Request.Builder();
    for (Map.Entry<String, String> entry : response.getActualSignedRequestHeaders().entrySet()) {
        builder.header(entry.getKey(), entry.getValue());
    }
    // 以下content为示例代码，需要通过列举已上传段方法的响应结果，拼装以下内容
    String content = "<CompleteMultipartUpload>";
    content += "<Part>";
    content += "<PartNumber>1</PartNumber>";
    content += "<ETag>da6a0d097e307ac52ed9b4ad551801fc</ETag>";
    content += "</Part>";
    content += "<Part>";
    content += "<PartNumber>2</PartNumber>";
    content += "<ETag>da6a0d097e307ac52ed9b4ad551801fc</ETag>";
    content += "</Part>";
    content += "</CompleteMultipartUpload>";
    // 使用POST请求合并段
    Request httpRequest =
        builder.url(response.getSignedUrl())
            .post(RequestBody.create(MediaType.parse(contentType), content.getBytes("UTF-8")))
            .build();
    OkHttpClient httpClient =
        new OkHttpClient.Builder()
            .followRedirects(false)
            .retryOnConnectionFailure(false)
            .cache(null)
            .build();
    Call c = httpClient.newCall(httpRequest);
    Response res = c.execute();
    System.out.println("\tStatus:" + res.code());
    if (res.body() != null) {
        System.out.println("\tContent:" + res.body().string() + "\n");
    }
    res.close();
} catch (ObsException e) {
    System.out.println("CompleteMultiPartUpload failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
}
```

```
        System.out.println("Host ID:" + e.getErrorHostId());
        e.printStackTrace();
    } catch (Exception e) {
        System.out.println("CompleteMultiPartUpload failed");
        // 其他异常信息打印
        e.printStackTrace();
    }
}
```

## 代码示例：获取图片转码的下载链接

临时授权访问，利用HttpMethodEnum.GET请求实现获取图片转码的下载链接，URL有效期设置3600秒。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import java.util.HashMap;
import java.util.Map;
public class GetObject002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以通过登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 获取图片转码的下载链接
            // URL有效期，3600秒
            long expireSeconds = 3600L;
            TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.GET,
            expireSeconds);
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname");
            // 设置图片转码参数
            Map<String, Object> queryParams = new HashMap<String, Object>();
            queryParams.put("x-image-process", "image/resize,m_fixed,w_100,h_100/rotate,100");
            request.setQueryParams(queryParams);
            TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
            // 获取支持图片转码的下载链接
            System.out.println("Getting object using temporary signature url:");
            System.out.println("SignedUrl:" + response.getSignedUrl());
        } catch (ObsException e) {
            System.out.println("getObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
        }
    }
}
```

```
        System.out.println("Request ID:" + e.getErrorRequestId());
        System.out.println("Host ID:" + e.getErrorHostId());
        e.printStackTrace();
    } catch (Exception e) {
        System.out.println("getObject failed");
        // 其他异常信息打印
        e.printStackTrace();
    }
}
}
```

## 代码示例：下载 SSE-C 加密类型的对象

临时授权访问，利用HttpMethodEnum.GET请求实现下载SSE-C加密类型的对象，URL有效期设置3600秒。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.HttpMethodEnum;
import com.obs.services.model.TemporarySignatureRequest;
import com.obs.services.model.TemporarySignatureResponse;
import okhttp3.Call;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
import java.util.HashMap;
import java.util.Map;
public class GetObject003 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 下载SSE-C加密类型的对象
            // URL有效期，3600秒
            long expireSeconds = 3600L;
            TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.GET,
            expireSeconds);
            request.setBucketName("examplebucket");
            request.setObjectKey("objectname");
            // 设置SSE-C加密方式的头域信息
            Map<String, String> headers = new HashMap<>();
            headers.put("x-obs-server-side-encryption-customer-algorithm", "AES256");
            // 设置加密使用的密钥，该头域由256-bit的密钥经过base64-encoded得到
            headers.put(
                "x-obs-server-side-encryption-customer-key",
                "your base64 sse-c key generated by AES-256 algorithm");
            // 设置加密使用的密钥的MD5值，该头域由密钥的128-bit MD5值经过base64-encoded得到
            headers.put("x-obs-server-side-encryption-customer-key-MD5", "the md5 value of your sse-c key");
            request.setHeaders(headers);
            TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
            System.out.println("Getting object using temporary signature url.");
        } catch (ObsException e) {
            System.out.println("getObject failed");
            e.printStackTrace();
        }
    }
}
```

```
System.out.println("\t" + response.getSignedUrl());
Request.Builder builder = new Request.Builder();
for (Map.Entry<String, String> entry : response.getActualSignedRequestHeaders().entrySet()) {
    builder.header(entry.getKey(), entry.getValue());
}
// 使用GET请求下载对象
Request httpRequest = builder.url(response.getSignedUrl()).get().build();
OkHttpClient httpClient =
    new OkHttpClient.Builder()
        .followRedirects(false)
        .retryOnConnectionFailure(false)
        .cache(null)
        .build();
Call c = httpClient.newCall(httpRequest);
Response res = c.execute();
System.out.println("Status:" + res.code());
if (res.body() != null) {
    System.out.println("Content:" + res.body().string() + "\n");
}
res.close();
System.out.println("getObject successfully");
} catch (ObsException e) {
    System.out.println("getObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

#### 📖 说明

- HttpMethodEnum是OBS Java SDK定义的枚举类型，代表请求方法类型。
- 加密密钥的计算方式，可以参考章节：[如何生成SSE-C方式的加密密钥](#)。

#### 相关链接

- 更多URL携带签名信息参考[url中携带签名](#)。
- 服务端加密接口返回的错误码含义、问题原因及处理措施可参考[OBS错误码](#)。

## 12.2 通过 Header 鉴权访问 OBS(Java SDK)

### 代码示例：上传对象

通过Header鉴权访问OBS，将“Hello OBS”字符串上传到桶examplebucket里的objectName中。

```
import com.obs.services.internal.Constants;
import com.obs.services.internal.utils.AbstractAuthentication;
import com.obs.services.model.AuthTypeEnum;
import shade.okhttp3.Call;
import shade.okhttp3.MediaType;
```



```
import shade.okhttp3.OkHttpClient;
import shade.okhttp3.Request;
import shade.okhttp3.RequestBody;
import shade.okhttp3.Response;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;
import java.util.TimeZone;
public class PutObjectTemporaryHeaderToObs {
    // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
    // 您可以通过登录访问管理控制台获取访问密钥AK/SK
    private static String ak = System.getenv("ACCESS_KEY_ID");
    private static String sk = System.getenv("SECRET_ACCESS_KEY_ID");
    // 【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/SecurityToken，也可以使用其他外部引入方式传入。
    private static String securityToken = System.getenv("SECURITY_TOKEN");
    // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
    //private static String endPoint = System.getenv("ENDPOINT");
    // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请根据实际情况填写。
    private static String endPoint = "obs.cn-north-4.myhuaweicloud.com";
    private static String bucketName = "examplebucket";
    private static String objectName = "objectName";
    private static String contentType = "application/xml";
    private static Map<String, String> headers = new HashMap<>();
    public static void main(String[] args) throws Exception {
        headers.put("Content-Type", contentType);
        DateFormat dateFormat = new SimpleDateFormat("E, dd MMM yyyy HH:mm:ss z", Locale.ENGLISH);
        dateFormat.setTimeZone(TimeZone.getTimeZone("GMT"));
        Date now = new Date(System.currentTimeMillis());
        headers.put("x-obs-date", dateFormat.format(now));
        sign("PUT", "/" + bucketName + "/" + objectName);
        Request.Builder builder = new Request.Builder();
        for (Map.Entry<String, String> entry : headers.entrySet()) {
            builder.header(entry.getKey(), entry.getValue());
        }
        String url = "https://" + bucketName + "." + endPoint + "/" + objectName;
        //使用PUT请求上传对象
        Request httpRequest = builder.url(url).put(RequestBody.create(MediaType.parse(contentType), "Hello
OBS".getBytes("UTF-8"))).build();
        OkHttpClient httpClient = new
OkHttpClient.Builder().followRedirects(false).retryOnConnectionFailure(false)
            .cache(null).build();
        Call c = httpClient.newCall(httpRequest);
        Response res = c.execute();
        System.out.println("\tStatus:" + res.code());
        if (res.body() != null) {
            System.out.println("\tContent:" + res.body().string() + "\n");
        }
        res.close();
    }
    public static void sign(String method, String url) throws Exception {
        AbstractAuthentication authentication =
            Constants.AUTHENTICATION_MAP.get(AuthTypeEnum.OBS);
        String stringToSign = authentication.makeServiceCanonicalString(method, url, headers, null,
            Constants.ALLOWED_RESOURCE_PARAMETER_NAMES);
        System.out.println("stringToSign: " + stringToSign);
        String auth = "OBS " + ak + ":" + AbstractAuthentication.calculateSignature(stringToSign, sk);
        headers.put("Authorization", auth);
    }
}
```

## 代码示例：初始化分段上传任务

通过Header鉴权访问OBS，初始化多段上传任务，上传的桶名为examplebucket，对象为objectName。

```
import com.obs.services.internal.Constants;
import com.obs.services.internal.utils.AbstractAuthentication;
import com.obs.services.model.AuthTypeEnum;
import com.obs.services.model.SpecialParamEnum;
import shade.okhttp3.Call;
import shade.okhttp3.MediaType;
import shade.okhttp3.OkHttpClient;
import shade.okhttp3.Request;
import shade.okhttp3.RequestBody;
import shade.okhttp3.Response;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;
import java.util.TimeZone;
public class initiateMultipartUpload {
    // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
    //private static String endPoint = System.getenv("ENDPOINT");
    // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
    private static String endPoint = "obs.cn-north-4.myhuaweicloud.com";
    private static String ak = System.getenv("ACCESS_KEY_ID");
    private static String sk = System.getenv("SECRET_ACCESS_KEY_ID");
    private static String bucketName = "examplebucket";
    private static String objectName = "objectName";
    private static String contentType = "application/xml";
    private static Map<String, String> headers = new HashMap<>();
    public static void main(String[] args) throws Exception {
        DateFormat dateFormat = new SimpleDateFormat("E, dd MMM yyyy HH:mm:ss z", Locale.ENGLISH);
        dateFormat.setTimeZone(TimeZone.getTimeZone("GMT"));
        Date now = new Date(System.currentTimeMillis());
        headers.put("x-obs-date", dateFormat.format(now));
        sign("POST", "/" + bucketName + "/" + objectName + "?uploads");
        Request.Builder builder = new Request.Builder();
        for (Map.Entry<String, String> entry : headers.entrySet()) {
            builder.header(entry.getKey(), entry.getValue());
        }
        String url = "https://" + bucketName + "." + endPoint + "/" + objectName + "?uploads";
        //使用POST请求初始化分段上传任务
        Request httpRequest = builder.url(url).post(RequestBody.create(null, "")).build();
        OkHttpClient httpClient = new
OkHttpClient.Builder().followRedirects(false).retryOnConnectionFailure(false)
        .cache(null).build();
        Call c = httpClient.newCall(httpRequest);
        Response res = c.execute();
        System.out.println("\tStatus:" + res.code());
        if (res.body() != null) {
            System.out.println("\tContent:" + res.body().string() + "\n");
        }
        res.close();
    }
    public static void sign(String method, String url) throws Exception {
        AbstractAuthentication authentication =
            Constants.AUTHENTICATION_MAP.get(AuthTypeEnum.OBS);
        String stringToSign = authentication.makeServiceCanonicalString(method, url, headers, null,
Constants.ALLOWED_RESOURCE_PARAMETER_NAMES);
        System.out.println("stringToSign: " + stringToSign);
        String auth = "OBS " + ak + ":" + AbstractAuthentication.calculateSignature(stringToSign, sk);
        headers.put("Authorization", auth);
    }
}
```

## 代码示例：上传段

通过Header鉴权访问OBS，将单段上传到桶examplebucket中的对象objectName。

```
import com.obs.services.internal.Constants;
import com.obs.services.internal.utils.AbstractAuthentication;
import com.obs.services.model.AuthTypeEnum;
import shade.okhttp3.Call;
import shade.okhttp3.MediaType;
import shade.okhttp3.OkHttpClient;
import shade.okhttp3.Request;
import shade.okhttp3.RequestBody;
import shade.okhttp3.Response;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;
import java.util.TimeZone;

public class UploadPart {
    // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
    //private static String endPoint = System.getenv("ENDPOINT");
    // endPoint填写桶所在的endpoint, 此处以华北-北京四为例，其他地区请按实际情况填写。
    private static String endPoint = "obs.cn-north-4.myhuaweicloud.com";
    private static String ak = System.getenv("ACCESS_KEY_ID");
    private static String sk = System.getenv("SECRET_ACCESS_KEY_ID");
    private static String bucketName = "examplebucket";
    private static String objectName = "objectName";
    private static String contentType = "application/xml";
    private static Map<String, String> headers = new HashMap<>();
    public static void main(String[] args) throws Exception {
        DateFormat dateFormat = new SimpleDateFormat("E, dd MMM yyyy HH:mm:ss z", Locale.ENGLISH);
        dateFormat.setTimeZone(TimeZone.getTimeZone("GMT"));
        Date now = new Date(System.currentTimeMillis());
        headers.put("x-obs-date", dateFormat.format(now));
        sign("PUT", "/" + bucketName + "/" + objectName + "?
partNumber=1&uploadId=0000018A0CA7D2144015626C08A6067A");
        Request.Builder builder = new Request.Builder();
        for (Map.Entry<String, String> entry : headers.entrySet()) {
            builder.header(entry.getKey(), entry.getValue());
        }
        String url = "https://" + bucketName + "." + endPoint + "/" + objectName + "?
partNumber=1&uploadId=0000018A0CA7D2144015626C08A6067A";
        //使用PUT请求上传对象
        Request httpRequest = builder.url(url).put(RequestBody.create(null, new byte[6 * 1024 *
1024])).build();
        OkHttpClient httpClient = new
OkHttpClient.Builder().followRedirects(false).retryOnConnectionFailure(false)
            .cache(null).build();
        Call c = httpClient.newCall(httpRequest);
        Response res = c.execute();
        System.out.println("\tStatus:" + res.code());
        if (res.body() != null) {
            System.out.println("\tContent:" + res.body().string() + "\n");
        }
        res.close();
    }
    public static void sign(String method, String url) throws Exception {
        AbstractAuthentication authentication =
            Constants.AUTHENTICATION_MAP.get(AuthTypeEnum.OBS);
        String stringToSign = authentication.makeServiceCanonicalString(method, url, headers, null,
            Constants.ALLOWED_RESOURCE_PARAMETER_NAMES);
        System.out.println("stringToSign: " + stringToSign);
        String auth = "OBS " + ak + ":" + AbstractAuthentication.calculateSignature(stringToSign, sk);
        headers.put("Authorization", auth);
    }
}
```

## 代码示例：合并段

通过Header鉴权访问OBS，实现多段上传时段的合并。

```
import com.obs.services.internal.Constants;
import com.obs.services.internal.utils.AbstractAuthentication;
import com.obs.services.model.AuthTypeEnum;
import shade.okhttp3.Call;
import shade.okhttp3.MediaType;
import shade.okhttp3.OkHttpClient;
import shade.okhttp3.Request;
import shade.okhttp3.RequestBody;
import shade.okhttp3.Response;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;
import java.util.TimeZone;
public class CompletePart {
    // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
    //private static String endPoint = System.getenv("ENDPOINT");
    // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
    private static String endPoint = "obs.cn-north-4.myhuaweicloud.com";
    private static String ak = System.getenv("ACCESS_KEY_ID");
    private static String sk = System.getenv("SECRET_ACCESS_KEY_ID");
    private static String bucketName = "examplebucket";
    private static String objectName = "objectName";
    private static String contentType = "application/xml";
    private static Map<String, String> headers = new HashMap<>();
    public static void main(String[] args) throws Exception {
        headers.put("Content-Type", contentType);
        DateFormat dateFormat = new SimpleDateFormat("E, dd MMM yyyy HH:mm:ss z", Locale.ENGLISH);
        dateFormat.setTimeZone(TimeZone.getTimeZone("GMT"));
        Date now = new Date(System.currentTimeMillis());
        headers.put("x-obs-date", dateFormat.format(now));
        sign("POST", "/" + bucketName + "/" + objectName + "?
uploadId=0000018A0CA7D2144015626C08A6067A");
        Request.Builder builder = new Request.Builder();
        for (Map.Entry<String, String> entry : headers.entrySet()) {
            builder.header(entry.getKey(), entry.getValue());
        }
        String url = "https://" + bucketName + "." + endPoint + "/" + objectName + "?
uploadId=0000018A0CA7D2144015626C08A6067A";
        //使用POST请求初始化分段上传任务
        // 以下content为示例代码，需要通过列举已上传段方法的响应结果，拼装以下内容
        String content = "<CompleteMultipartUpload>";
        content += "<Part>";
        content += "<PartNumber>1</PartNumber>";
        content += "<ETag>da6a0d097e307ac52ed9b4ad551801fc</ETag>";
        content += "</Part>";
        content += "</CompleteMultipartUpload>";
        Request httpRequest = builder.url(url).post(RequestBody.create(MediaType.parse(contentType),
content.getBytes("UTF-8"))).build();
        OkHttpClient httpClient = new
OkHttpClient.Builder().followRedirects(false).retryOnConnectionFailure(false)
.cache(null).build();
        Call c = httpClient.newCall(httpRequest);
        Response res = c.execute();
        System.out.println("\tStatus:" + res.code());
        if (res.body() != null) {
            System.out.println("\tContent:" + res.body().string() + "\n");
        }
        res.close();
    }
    public static void sign(String method, String url) throws Exception {
        AbstractAuthentication authentication =
Constants.AUTHENTICATION_MAP.get(AuthTypeEnum.OBS);
```

```
String stringToSign = authentication.makeServiceCanonicalString(method, url, headers, null,
Constants.ALLOWED_RESOURCE_PARAMETER_NAMES);
System.out.println("stringToSign: " + stringToSign);
String auth = "OBS " + ak + ":" + AbstractAuthentication.calculateSignature(stringToSign, sk);
headers.put("Authorization", auth);
}
}
```

## 示例代码：下载对象

通过Header鉴权访问OBS，下载examplebucket里的objectName对象。

```
import com.obs.services.internal.Constants;
import com.obs.services.internal.utils.AbstractAuthentication;
import com.obs.services.model.AuthTypeEnum;
import shade.okhttp3.Call;
import shade.okhttp3.OkHttpClient;
import shade.okhttp3.Request;
import shade.okhttp3.Response;
import java.io.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.time.ZoneId;
import java.time.ZoneOffset;
import java.util.*;

public class Header_Get_Object_test {
    private static String ak = System.getenv("ACCESS_KEY_ID");
    private static String sk = System.getenv("SECRET_ACCESS_KEY_ID");
    // 【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/SecurityToken，也可以使用其他外部引入方式传入。
    private static String securityToken = System.getenv("SECURITY_TOKEN");
    // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
    //private static String endPoint = System.getenv("ENDPOINT");
    // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
    private static String endPoint = "obs.cn-north-4.myhuaweicloud.com";
    private static String bucketName = "examplebucket" ;
    private static String objectName = "objectkname";
    private static String contentType = "application/xml";
    private static String method = "GET";
    private static Map<String, String> headers = new HashMap<>();

    public static void main(String[] args) throws Exception {
        headers.put("Content-Type", contentType);
        DateFormat dateFormat = new SimpleDateFormat("E, dd MMM yyyy HH:mm:ss z", Locale.ENGLISH);
        ZoneId zoneId = ZoneId.ofOffset("UTC", ZoneOffset.ofHours(8));
        TimeZone timeZone = TimeZone.getTimeZone(zoneId);
        dateFormat.setTimeZone(timeZone);
        String now = dateFormat.format(new Date(System.currentTimeMillis()));
        headers.put("x-obs-date", now);
        sign(method, "/" + bucketName + "/" + objectName);
        Request.Builder builder = new Request.Builder();
        for (Map.Entry<String, String> entry : headers.entrySet()) {
            builder.header(entry.getKey(), entry.getValue());
        }
        String url = "https://" + bucketName + "." + endPoint + "/" + objectName ;
        //使用get请求下载对象
        Request httpRequest = builder.url(url).get().build();
        OkHttpClient httpClient =
            new OkHttpClient.Builder()
                .followRedirects(false)
                .retryOnConnectionFailure(false)
                .cache(null)
                .build();
        Call c = httpClient.newCall(httpRequest);
        Response res = c.execute();
        if (res.code() < 300)
```

```
        System.out.println("Status:" + res.code());
        System.out.println("getObject successfully");
        InputStream objectContent = null;
        if (res.body() != null) {
            objectContent = res.body().byteStream();
            System.out.println("Content:" + res.body().string() + "\n");
        }
        if(objectContent != null) {
            // objectContent 就是获取的要下载的文件的文件流
            // 在这里可以读取 objectContent, 长时间不读取这个流会被服务端断开连接
        }
        res.close();
    }
    public static void sign(String method, String url) throws Exception {
        AbstractAuthentication authentication =
            Constants.AUTHENTICATION_MAP.get(AuthTypeEnum.OBS);
        String stringToSign = authentication.makeServiceCanonicalString(method, url, headers, null,
            Constants.ALLOWED_RESOURCE_PARAMETER_NAMES);
        System.out.println("stringToSign: " + stringToSign);
        String auth = "OBS " + ak + ":" + AbstractAuthentication.calculateSignature(stringToSign, sk);
        headers.put("Authorization", auth);
    }
}
```

# 13 多版本控制(Java SDK)

## 13.1 多版本控制简介(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

OBS支持保存一个对象的多个版本，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。

更多关于多版本控制的内容请参见[多版本控制](#)。

## 13.2 设置桶多版本状态(Java SDK)

### 功能说明

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。

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 您必须是桶拥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以及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(final [SetBucketVersioningRequest request](#))

## 请求参数说明

表 13-1 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">SetBucketVersioningRequest</a>	必选	<b>参数解释：</b> 设置桶多版本状态请求参数，详见 <a href="#">SetBucketVersioningRequest</a> 。



表 13-2 SetBucketVersioningRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
status	<a href="#">VersioningStatusEnum</a>	必选	<p><b>参数解释:</b> 桶的多版本状态。</p> <p><b>取值范围:</b> 详见<a href="#">VersioningStatusEnum</a>。</p>

表 13-3 VersioningStatusEnum

常量名	原始值	说明
SUSPENDED	Suspended	暂停多版本。
ENABLED	Enabled	开启多版本。

## 返回结果说明

表 13-4 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

以下代码展示了如何通过启用和暂停设置examplebucket桶的多版本状态。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketVersioningConfiguration;
import com.obs.services.model.VersioningStatusEnum;
public class SetBucketVersioning001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 启用桶多版本状态
            obsClient.setBucketVersioning(
```

```
        "examplebucket", new BucketVersioningConfiguration(VersioningStatusEnum.ENABLED));
    // 暂停桶多版本状态
    obsClient.setBucketVersioning(
        "examplebucket", new BucketVersioningConfiguration(VersioningStatusEnum.SUSPENDED));
    System.out.println("setBucketVersioning successfully");
} catch (ObsException e) {
    System.out.println("setBucketVersioning failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("setBucketVersioning failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置桶的多版本状态的API说明，请参见[设置桶的多版本状态](#)。
- 更多关于设置桶的多版本状态的示例代码，请参见[Github示例](#)。
- 设置桶的多版本状态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](#)。
- 多版本控制相关问题请参见[多版本控制相关常见问题](#)。

## 13.3 获取桶多版本状态(Java SDK)

### 功能说明

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以及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(final [BaseBucketRequest request](#))

## 请求参数说明

表 13-5 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">BaseBucketRequest</a>	必选	<b>参数解释：</b> 查看桶多版本状态请求参数，详见 <a href="#">BaseBucketRequest</a> 。

表 13-6 BaseBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my.bucket”和“my.bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值：</b> 无</p>

## 返回结果说明

表 13-7 BucketVersioningConfiguration

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
status	<a href="#">VersioningStatusEnum</a>	<p><b>参数解释:</b> 桶的多版本状态。</p> <p><b>取值范围:</b> 详见<a href="#">VersioningStatusEnum</a>。</p>

表 13-8 VersioningStatusEnum

常量名	原始值	说明
SUSPENDED	Suspended	暂停多版本。
ENABLED	Enabled	开启多版本。

## 代码示例

以下代码展示了如何查看examplebucket桶的多版本状态。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketVersioningConfiguration;
public class GetBucketVersioning001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
```

```
// 【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/SecurityToken，也可以使用其他外部引入方式传入。
String securityToken = System.getenv("SECURITY_TOKEN");
// endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 查看桶多版本状态
    BucketVersioningConfiguration status = obsClient.getBucketVersioning("examplebucket");
    System.out.println("getBucketVersioning successfully");
    System.out.println("getVersioningStatus:" + status.getVersioningStatus());
} catch (ObsException e) {
    System.out.println("getBucketVersioning failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getBucketVersioning failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于查看桶多版本状态的API说明，请参见[获取桶的多版本状态](#)。
- 更多关于查看桶多版本状态的示例代码，请参见[Github示例](#)。
- 查看桶多版本状态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](#)。
- 多版本控制相关问题请参见[多版本控制相关常见问题](#)。

## 13.4 获取多版本对象(Java SDK)

### 功能说明

您可以通过ObsClient.getObject接口传入版本号（versionId）来获取桶中指定版本的对象。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 接口约束

- 您必须是桶所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以及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([GetObjectRequest request](#))

## 请求参数说明

表 13-9 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">GetObjectRequest</a>	必选	<b>参数解释：</b> 桶下载对象时请求参数，详见 <a href="#">GetObjectRequest</a> 。

表 13-10 GetObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
versionId	String	可选	<b>参数解释:</b> 对象的版本号。版本号为空，默认下载最新版本的对象。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无。
imageProcess	String	可选	<b>参数解释:</b> 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。 <b>取值范围:</b> 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 <a href="#">处理图片</a> 。 <b>默认取值:</b> 如果不输入处理命令，将返回原图。
rangeStart	Long	可选	<b>参数解释:</b> 指定下载对象的起始位置。 <b>取值范围:</b> 非负整数。 <b>默认取值:</b> 无
rangeEnd	Long	可选	<b>参数解释:</b> 指定下载对象的结束位置。 <b>约束限制:</b> 如果该值大于对象长度-1，实际仍取对象长度-1。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
ifMatchTag	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
ifNoneMatchTag	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
IfModifiedSince	Date	可选	<p><b>参数解释:</b> 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, 否则返回错误。</p> <p><b>默认取值:</b> 无</p>
IfUnmodifiedSince	Date	可选	<p><b>参数解释:</b> 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, 否则返回错误。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端解密头信息, 详见<a href="#">SseCHeader</a>。</p>
replaceMetadata	<a href="#">ObjectReplaceMetadata</a>	可选	<p><b>参数解释:</b> 指定OBS返回请求的相关信息, 详见<a href="#">ObjectReplaceMetadata</a>。</p>
progressListener	<a href="#">ProgressListener</a>	可选	<p><b>参数解释:</b> 设置数据传输监听器, 用于获取下载进度, 详见<a href="#">ProgressListener</a>。</p>

表 13-11 ObjectRepleaceMetadata

参数名称	参数类型	是否必选	描述
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 无</p>
contentType	String	可选	<p><b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p> <p><b>默认取值:</b> 无</p>
expires	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示或在访问时弹出文件下载对话框。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
contentEncoding	String	可选	<b>参数解释:</b> 响应中的Content-Encoding头。指定对象被下载时的内容编码格式。 <b>默认取值:</b> 无

表 13-12 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<b>参数解释:</b> 获取上传进度，详见 <a href="#">progressChanged</a> 。 <b>默认取值:</b> 无

表 13-13 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<b>参数解释:</b> 传输进度数据，详见 <a href="#">ProgressStatus</a> 。 <b>默认取值:</b> 无

表 13-14 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。

方法名称	返回值类型	说明
getTransferredBytes() ( )	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 13-15 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard, AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥，byte[]形式，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥，由原始密钥经过Base64编码后得到，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 13-16 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 13-17 ServerAlgorithm

常量值	原始值
AES256	AES256

表 13-18 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 13-19 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 13-20 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 单次上传对象大小范围是[0, 5GB]。</li> <li>• 如果需要上传超过5GB的大文件，需要通过<a href="#">多段操作</a>来分段上传。</li> </ul> <p><b>默认取值:</b> 如果不设置，则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？（Java SDK）</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无
contentType	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无
contentTypeMD5	String	可选	<b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。 <b>约束限制:</b> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <b>取值范围:</b> 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：n58IG6hfM7vqI4K0vnWpog== <b>默认取值:</b> 无



参数名称	参数类型	是否必选	描述
storageClass	StorageClassEnum	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时,可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>
websiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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,该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>必须以“/”、“http://”或“https://”开头,长度不超过2KB。</li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度,单位:字节。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称, Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取, 参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 一个对象可以有多个元数据, 总大小不能超过8KB。</li><li>• 使用ObsClient.getObject下载对象时, 对象的自定义元数据也会同时下载。</li></ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 13-21 ObsObject

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 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.cn- north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
owner	<b>Owner</b>	<p><b>参数解释:</b> 所有者，详见<b>Owner</b>。</p>
metadata	<b>ObjectMetadata</b>	<p><b>参数解释:</b> 对象元数据，详见<b>ObjectMetadata</b>。</p>

参数名称	参数类型	描述
objectContent	InputStream	<b>参数解释:</b> 对象数据流。 <b>默认取值:</b> 无

## 代码示例

以下代码展示了如何通过设置versionId获取多版本对象查看examplebucket桶内对象objectname的多版本状态。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObsObject;
public class GetObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 设置versionId获取多版本对象
            ObsObject obsObject = obsClient.getObject("examplebucket", "objectname", "versionid1");
            System.out.println("getObject successfully");
            obsObject.getObjectContent().close();
        } catch (ObsException e) {
            System.out.println("getObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("getObject failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于下载多版本对象的API说明，请参见[获取多版本对象内容](#)。
- 下载对象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](#)。
- 下载多版本对象常见问题请参见[下载对象失败](#)。

## 13.5 复制多版本对象(Java SDK)

### 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。

您可以通过ObsClient.copyObject接口传入版本号（versionId）来复制多版本对象。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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](#)。
- 用户有待复制的源对象的读权限。
- OBS支持的region以及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拷贝的原桶和对象信息，不能携带消息实体。
- 支持同区域跨桶复制，不支持跨区域复制。
- 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[分段复制](#)功能复制对象。
- 如果待复制的源对象是归档存储类别，则必须先恢复源对象才能进行复制。

### 方法定义

obsClient.copyObject([CopyObjectRequest request](#))

## 请求参数说明

表 13-22 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">CopyObjectRequest</a>	必选	<b>参数解释：</b> 复制对象请求参数，详见 <a href="#">CopyObjectRequest</a> 。

表 13-23 CopyObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释：</b> 目标桶名。 <b>约束限制：</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <b>默认取值：</b> 无

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 目标对象名。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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
sseKmsHeader	<a href="#">SseKmsHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息，用于加密目标对象。详见<a href="#">SseKmsHeader</a>。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息，用于加密目标对象。详见<a href="#">SseCHeader</a>。</p> <p><b>默认取值:</b> 无</p>
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 复制对象时可指定的预定义访问策略。详见<a href="#">AccessControlList</a>。</p> <p><b>默认取值:</b> 无</p>
successRedirectLocation	String	可选	<p><b>参数解释:</b> 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。</p> <ul style="list-style-type: none"> <li>如果此参数值有效且操作成功，响应码为303，返回值中的Location头域由此参数以及桶名、对象名、对象的ETag组成。</li> <li>如果此参数值无效，则OBS忽略此参数的作用，返回值中的Location头域为对象地址，响应码根据操作成功或失败正常返回。</li> </ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sourceBucket Name	String	必选	<p><b>参数解释:</b> 源桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
sourceObject Key	String	必选	<p><b>参数解释:</b> 源对象名。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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
newObjectMetadata	<a href="#">ObjectMetadata</a>	可选	<p><b>参数解释:</b> 目标对象的自定义元数据，详见<a href="#">ObjectMetadata</a>。</p> <p><b>约束限制:</b> newObjectMetadata需与replaceMetadata配合使用。</p> <p><b>默认取值:</b> 无</p>
replaceMetadata	boolean	可选	<p><b>参数解释:</b> 复制对象时，是否重写源对象的元数据。</p> <p><b>约束限制:</b> replaceMetadata需与newObjectMetadata配合使用。</p> <p><b>取值范围:</b> true: 重写源对象的元数据。 false: 不重写源对象的元数据。</p> <p><b>默认取值:</b> 无</p>
ifModifiedSince	java.util.Date	可选	<p><b>参数解释:</b> 如果源对象的修改时间早于该参数值指定的时间，则进行复制，否则返回错误。</p> <p><b>默认取值:</b> 无</p>
ifUnmodifiedSince	java.util.Date	可选	<p><b>参数解释:</b> 如果源对象的修改时间晚于该参数值指定的时间，则进行复制，否则返回错误。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
ifMatchTag	String	可选	<p><b>参数解释:</b> 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。其中, 源对象的ETag值是指源对象数据的MD5校验值。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 如果包含ifUnmodifiedSince并且不符合, 或者包含ifMatchTag并且不符合, 或者包含ifModifiedSince并且不符合, 或者包含ifNoneMatchTag并且不符合, 则复制失败, 抛出异常中HTTP状态码为: 412 precondition failed。</li> <li>• ifModifiedSince和ifNoneMatchTag可以一起使用; ifUnmodifiedSince和ifMatchTag可以一起使用。</li> </ul> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
ifNoneMatchTag	String	可选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 源对象版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
sseCHeaderSource	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端解密头信息, 用于解密源对象。详见<a href="#">SseCHeader</a>。</p> <p><b>默认取值:</b> 无</p>

表 13-24 SseKmsHeader

参数名称	参数类型	是否必选	描述
encryption	<a href="#">ServerEncryption</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p><b>取值范围:</b> 可选值: kms, 即选择SSE-KMS方式加密对象, 详见<a href="#">ServerEncryption</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持kms。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
kmsKeyId	String	可选	<p><b>参数解释:</b> SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p><b>取值范围:</b> 有效值支持两种格式:</p> <ol style="list-style-type: none"> <li>regionID:domainID(账号ID):key/key_id</li> <li>key_id</li> </ol> <p>其中:</p> <ul style="list-style-type: none"> <li>regionID是使用密钥所属region的ID, 可在<a href="#">地区和终端节点</a>页面获取;</li> <li>domainID是使用密钥所属账号的账号ID, 获取方法参见<a href="#">如何获取账号ID和用户ID?</a>;</li> <li>key_id是从数据加密服务创建的密钥ID, 获取方法请参见<a href="#">查看密钥</a>。</li> </ul> <p><b>默认取值:</b></p> <ul style="list-style-type: none"> <li>如果用户没有提供该头域, 那么默认的主密钥将会被使用。</li> <li>如果默认主密钥不存在, 将默认创建并使用。</li> </ul>

表 13-25 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard, AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥，byte[]形式，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥，由原始密钥经过Base64编码后得到，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 13-26 ServerEncryption

常量值	原始值
OBS_KMS	kms

表 13-27 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 13-28 ServerAlgorithm

常量值	原始值
AES256	AES256

表 13-29 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	<b>Owner</b>	可选	<b>参数解释:</b> 桶所有者的信息, 详见 <b>Owner</b> 。
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false
grants	Set< <b>GrantAndPermission</b> >	可选	<b>参数解释:</b> 被授权用户相关信息, 详见 <b>GrantAndPermission</b> 。

表 13-30 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<p><b>参数解释:</b> 桶所有者的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
display Name	String	可选	<p><b>参数解释:</b> 所有者的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 13-31 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<p><b>参数解释:</b> 被授权用户或用户组, 详见<a href="#">GranteeInterface</a>。</p>
permission	<a href="#">Permission</a>	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 13-32 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。
PERMISSION_WRITE_ACP	WRITE_ACP	修改ACL配置的权限。 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。
PERMISSION_FULL_CONTROL	FULL_CONTROL	完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。 如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。 如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。

表 13-33 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 13-34 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 13-35 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。



表 13-36 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 单次上传对象大小范围是[0, 5GB]。</li> <li>• 如果需要上传超过5GB的大文件，需要通过<a href="#">多段操作</a>来分段上传。</li> </ul> <p><b>默认取值:</b> 如果不设置，则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无
contentType	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无
contentTypeMD5	String	可选	<b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。 <b>约束限制:</b> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <b>取值范围:</b> 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== <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
storageClass	StorageClassEnum	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>
websiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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，该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头，长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度，单位：字节。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称, Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取, 参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据, 总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时, 对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 13-37 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

## 返回结果说明

表 13-38 CopyObjectResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
lastModified	java.util.Date	<b>参数解释:</b> 目标对象的最近一次修改时间。 <b>默认取值:</b> 无
etag	String	<b>参数解释:</b> 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。上传的对象或拷贝操作创建的对象，都有唯一的ETag。 <b>约束限制:</b> 当对象是服务端加密的对象时，ETag值不是对象的MD5值。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	描述
versionId	String	<b>参数解释:</b> 目标对象的版本号。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
copySourceVersionId	String	<b>参数解释:</b> 源对象的版本号。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
storageClass	<a href="#">StorageClassEnum</a>	<b>参数解释:</b> 目标对象的存储类别。 <b>取值范围:</b> 详见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无

## 代码示例

以下代码展示了如何通过设置versionId复制sourceexamplebucket桶中sourceobjectname的多版本对象。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.CopyObjectRequest;
public class CopyObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
```

```
// 复制多版本对象
CopyObjectRequest request = new CopyObjectRequest();
request.setSourceBucketName("sourceexamplebucket");
request.setSourceObjectKey("sourceobjectname");
// 设置要复制对象的版本号
request.setVersionId("versionid");
request.setDestinationBucketName("destexamplebucket");
request.setDestinationObjectKey("destobjectname");
obsClient.copyObject(request);
System.out.println("copyObject successfully");
} catch (ObsException e) {
    System.out.println("copyObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("copyObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于复制对象的API说明, 请参见[复制对象](#)。
- 复制对象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](#)。
- 复制对象常见问题请参见[我可以在桶间进行文件复制吗?](#)。

## 13.6 恢复多版本归档存储对象(Java SDK)

### 功能说明

如果要下载归档存储对象, 需要先将归档存储对象恢复。恢复归档存储对象的恢复选项可支持两类, 见[RestoreTierEnum](#):

您可以通过ObsClient.restoreObject接口传入版本号 (versionId) 来恢复多版本归档存储对象。

#### 须知

开发过程中, 您有任何问题可以在github上[提交issue](#), 或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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](#)。
- ObsClient.restoreObject中指定的对象必须是归档存储类别，否则调用该接口会抛出异常。

 **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。

## 方法定义

obsClient.restoreObject([RestoreObjectRequest request](#))

## 请求参数说明

表 13-39 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">RestoreObjectRequest</a>	必选	<b>参数解释：</b> 恢复多版本归档存储对象请求参数，详见 <a href="#">RestoreObjectRequest</a> 。



表 13-40 RestoreObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
versionId	String	可选	<b>参数解释:</b> 待恢复归档存储对象的版本号。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无, 如果不设置则默认指定最新版本的对象。
days	int	必选	<b>参数解释:</b> 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。 <b>约束限制:</b> 取值必须是正整数。 <b>取值范围:</b> [1, 30], 单位: 天。 <b>默认取值:</b> 无
tier	<a href="#">RestoreTierEnum</a>	可选	<b>参数解释:</b> 恢复选项, 表示恢复对象所耗的时间。 <b>取值范围:</b> 详见 <a href="#">RestoreTierEnum</a> 。 <b>默认取值:</b> 默认取值为标准恢复。

表 13-41 RestoreTierEnum

可选值	原始值	描述
EXPEDITED	Expedited	快速恢复, 恢复耗时1~5分钟。
STANDARD	Standard	标准恢复, 恢复耗时3~5小时。

## 返回结果说明

表 13-42 RestoreObjectStatus

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

使用快速恢复方式，恢复多版本对象，其中桶名为examplebucket，桶中对象为objectname，保留时间为1天。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ObsObject;
import com.obs.services.model.RestoreObjectRequest;
import com.obs.services.model.RestoreTierEnum;
public class RestoreObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 使用快速恢复方式，恢复多版本对象
            RestoreObjectRequest request = new RestoreObjectRequest("examplebucket", "objectname", 1);
```

```
request.setRestoreTier(RestoreTierEnum.EXPEDITED);
request.setVersionId("versionid");
obsClient.restoreObject(request);
System.out.println("RestoreObject successfully");
} catch (ObsException e) {
    System.out.println("RestoreObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("RestoreObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于恢复归档存储对象的API说明, 请参见[恢复归档存储对象](#)。
- 更多关于恢复归档存储对象的示例代码, 请参见[Github示例](#)。
- 恢复归档存储对象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](#)。

## 13.7 列举多版本对象(Java SDK)

### 功能说明

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多版本对象信息。返回结果以多版本对象名的字典序排序。

ObsClient.listVersions返回结果包含多版本对象和对象删除标记。

### 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以及region与endPoint的对应关系, 详细信息请参见[地区与终端节点](#)。
- 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, 可以采用[分页列举](#)的方式返回桶中所有多版本对象。列举时, ListVersionsResult.isTruncated为true表明本次没有返回全部对象, 可以通过ListVersionsResult.getNextKeyMarker和ListVersionsResult.getNextVersionIdMarker获取下次列举的起始位置。

## 方法定义

obsClient.listVersions([ListVersionsRequest request](#))

## 请求参数说明

表 13-43 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">ListVersionsRequest</a>	必选	<b>参数解释:</b> 列举多版本对象请求参数, 详见 <a href="#">ListVersionsRequest</a> 。

表 13-44 请求参数 ListVersionsRequest 成员说明

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释:</b> 桶名。 <b>约束限制:</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下:<ul style="list-style-type: none"><li>3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻(如: “my.bucket”)。</li><li>禁止“.”和“-”相邻(如: “my-.bucket”和“my.-bucket”)。</li></ul></li><li>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。</li></ul> <b>默认取值:</b> 无
prefix	String	可选	<b>参数解释:</b> 限定返回的对象名必须带有prefix前缀。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
keyMarker	String	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 该字段仅用于多版本列举。</p> <p><b>取值范围:</b> 上次请求返回体的nextKeyMarker值。</p> <p><b>默认取值:</b> 无</p>
maxKeys	int	可选	<p><b>参数解释:</b> 列举多版本对象的最大数目，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><b>取值范围:</b> 1~1000，当超出范围时，按照默认的1000进行处理。</p> <p><b>默认取值:</b> 1000</p>

参数名称	参数类型	是否必选	描述
delimiter	String	可选	<p><b>参数解释:</b> 将对象名进行分组的分隔符。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 commonPrefix；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 commonPrefix，而 bbcde 会被单独分成一组，形成一条前缀为 bbcde 的 commonPrefix。</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><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId Marker	String	可选	<p><b>参数解释:</b> 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><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>该字段只适用于多版本列举场景。</li> <li>如果 versionIdMarker 不是 keyMarker 的一个版本号，则该参数无效。</li> </ul> <p><b>取值范围:</b> 对象的版本号，即上次请求返回体的 nextVersionIdMarker 值。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
encodingType	String	可选	<p><b>参数解释:</b> 对响应中的部分元素进行指定类型的编码。如果 delimiter、keyMarker、prefix、nextKeyMarker 和 key 包含 xml 1.0 标准不支持的控制字符, 可通过设置 encodingType 对响应中的 delimiter、keyMarker、prefix (包括 commonPrefixes 中的 prefix)、nextKeyMarker 和 key 进行编码。</p> <p><b>取值范围:</b> 可选值为 url。</p> <p><b>默认取值:</b> 无, 不设置则不编码。</p>

## 返回结果说明

表 13-45 ListVersionsResult

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下: <ul style="list-style-type: none"> <li>3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻(如: “my..bucket”)。</li> <li>禁止“.”和“-”相邻(如: “my-.bucket”和“my.-bucket”)。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	描述
prefix	String	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionIdMarker	String	<p><b>参数解释:</b> 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</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>该字段只适用于多版本列举场景。</li> <li>如果versionIdMarker不是keyMarker的一个版本号，则该参数无效。</li> </ul> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
nextVersionIdMarker	String	<p><b>参数解释:</b> 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> <p><b>约束限制:</b> 该字段只适用于多版本列举场景。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
objectSummaries	List<ObsObject>	<p><b>参数解释:</b> 对象信息，详见<a href="#">ObsObject</a>。</p>

参数名称	参数类型	描述
commonPrefixes	List<String>	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
truncated	boolean	<p><b>参数解释:</b> 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><b>取值范围:</b> true: 表示没有返回全部结果。 false: 表示已返回了全部结果。</p> <p><b>默认取值:</b> 无</p>
keyMarker	String	<p><b>参数解释:</b> 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。</p> <p><b>约束限制:</b> 该字段仅用于多版本列举。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
maxKeys	int	<p><b>参数解释:</b> 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> <p><b>取值范围:</b> 1~1000，当超出范围时，按照默认的1000进行处理。</p> <p><b>默认取值:</b> 1000</p>

参数名称	参数类型	描述
delimiter	String	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
nextKeyMarker	String	<p><b>参数解释:</b> 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> <p><b>约束限制:</b> 该字段仅用于多版本列举。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
location	String	<p><b>参数解释:</b> 桶所在的区域。</p> <p><b>约束限制:</b> 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><b>取值范围:</b> 当前有效的OBS区域和终端节点的更多信息，请参考<a href="#">地区和终端节点</a>。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><b>默认取值:</b> 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>
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
versions	<a href="#">VersionOrDeleteMarker</a> []	<p><b>参数解释:</b> 对象多版本信息，详见<a href="#">VersionOrDeleteMarker</a>。</p>

表 13-46 ObsObject

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
owner	<b>Owner</b>	<p><b>参数解释:</b> 所有者，详见<b>Owner</b>。</p>
metadata	<b>ObjectMetadata</b>	<p><b>参数解释:</b> 对象元数据，详见<b>ObjectMetadata</b>。</p>
objectContent	InputStream	<p><b>参数解释:</b> 对象数据流。</p> <p><b>默认取值:</b> 无</p>

表 13-47 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 13-48 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<b>参数解释:</b> 对象数据的长度。 <b>约束限制:</b> <ul style="list-style-type: none"><li>• 单次上传对象大小范围是[0, 5GB]。</li><li>• 如果需要上传超过5GB的大文件, 需要通过<a href="#">多段操作</a>来分段上传。</li></ul> <b>默认取值:</b> 如果不设置, 则SDK会自动计算对象数据的长度。

参数名称	参数类型	是否必选	描述
contentType	String	可选	<b>参数解释:</b> 对象的文件类型 ( MIME类型 )。contentType ( 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。 <b>取值范围:</b> 常见的contentType ( MIME ) 列表参见 <a href="#">如何理解Content-Type ( MIME ) ? (Java SDK)</a> 。 <b>默认取值:</b> 如果未指定Content-Type, SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type ( 如.xml判断为application/xml文件; .html判断为text/html文件 )。
contentEncoding	String	可选	<b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。 <b>默认取值:</b> 无
contentDisposition	String	可选	<b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象, 当下载对象或者访问对象时, 以默认文件名命名的文件将直接在浏览器上显示, 或在弹出文件下载对话框时显示。 <b>默认取值:</b> 无
cacheControl	String	可选	<b>参数解释:</b> 响应中的Cache-Control头。指定对象被下载时的网页的缓存行为。 <b>默认取值:</b> 无
contentTypeLanguage	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合, 以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
expires	String	可选	<p><b>参数解释:</b> 指对象在网页中的缓存过期时间。</p> <p><b>约束限制:</b> 日期格式为GMT的格式。</p> <p><b>默认取值:</b> 无</p>
contentMd5	String	可选	<p><b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>对象数据的MD5值必须经过Base64编码。</li> <li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li> </ul> <p><b>取值范围:</b> 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><b>默认取值:</b> 无</p>
storageClasses	<a href="#">StorageClassEnum</a>	可选	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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, 该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头, 长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度, 单位: 字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是, 可追加上传。 false: 否, 不可追加上传。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 13-49 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 13-50 VersionOrDeleteMarker

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my.bucket”和“my.bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
key	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
isLatest	boolean	<b>参数解释:</b> 标识对象是否是最新的版本。 <b>取值范围:</b> <ul style="list-style-type: none"><li>• true: 代表是最新的版本。</li><li>• false: 代表非最新版本。</li></ul> <b>默认取值:</b> false
lastModified	Date	<b>参数解释:</b> 对象最近一次被修改的时间 (UTC时间)。 <b>取值范围:</b> UTC时间 <b>默认取值:</b> 无
owner	<b>Owner</b>	<b>参数解释:</b> 用户信息, 包含对象所有者DomainId和对象拥有者名称, 详见 <b>Owner</b> 。
etag	String	<b>参数解释:</b> 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。实际标签是对象的哈希。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。上传的对象或拷贝操作创建的对象, 通过MD5加密后都有唯一的ETag。 <b>约束限制:</b> 当对象是服务端加密的对象时, ETag值不是对象的MD5值。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
size	long	<b>参数解释:</b> 对象的字节数。 <b>取值范围:</b> 0~48.8TB, 单位: 字节。 <b>默认取值:</b> 无

参数名称	参数类型	描述
storageClass	<b>StorageClassEnum</b>	<p><b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。</p> <p><b>取值范围:</b> 可选择的存储类别参见<b>StorageClassEnum</b>。</p> <p><b>默认取值:</b> 无</p>
isDeleteMarker	boolean	<p><b>参数解释:</b> 是否为删除标记。</p> <p><b>取值范围:</b> true: 为删除标记。 false: 不为删除标记。</p> <p><b>默认取值:</b> 无</p>

## 代码示例：简单列举多版本对象

以下代码展示如何简单列举examplebucket桶中的多版本对象，最多返回1000个对象。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListVersionsResult;
import com.obs.services.model.VersionOrDeleteMarker;
public class ListVersions001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 简单列举
            ListVersionsResult result = obsClient.listVersions("examplebucket");
            System.out.println("listVersions successfully");
            for (VersionOrDeleteMarker v : result.getVersions()) {
                System.out.println("Key:" + v.getKey());
            }
        }
    }
}
```

```
        System.out.println("Owner:" + v.getOwner());
        System.out.println("isDeleteMarker:" + v.isDeleteMarker());
    }
} catch (ObsException e) {
    System.out.println("listVersions failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("listVersions failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：指定数目列举多版本对象

以下代码展示如何向examplebucket桶中指定数目列举多版本对象。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListVersionsResult;
import com.obs.services.model.VersionOrDeleteMarker;
public class ListVersions002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 指定数目列举
            ListVersionsResult result = obsClient.listVersions("examplebucket", 100);
            System.out.println("listVersions successfully");
            for (VersionOrDeleteMarker v : result.getVersions()) {
                System.out.println("Key:" + v.getKey());
                System.out.println("Owner:" + v.getOwner());
                System.out.println("isDeleteMarker:" + v.isDeleteMarker());
            }
        } catch (ObsException e) {
            System.out.println("listVersions failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
        }
    }
}
```

```
System.out.println("Error Message:" + e.getMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("listVersions failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：指定前缀列举多版本对象

以下代码展示如何指定带有prefix前缀列举examplebucket桶中的多版本对象，最大数目为100个。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListVersionsRequest;
import com.obs.services.model.ListVersionsResult;
import com.obs.services.model.VersionOrDeleteMarker;
public class ListVersions003 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 指定前缀列举
            // 列举带有prefix前缀的100个多版本对象
            ListVersionsRequest request = new ListVersionsRequest("examplebucket", 100);
            request.setPrefix("prefix");
            ListVersionsResult result = obsClient.listVersions(request);
            System.out.println("listVersions successfully");
            for (VersionOrDeleteMarker v : result.getVersions()) {
                System.out.println("Key:" + v.getKey());
                System.out.println("Owner:" + v.getOwner());
                System.out.println("isDeleteMarker:" + v.isDeleteMarker());
            }
        } catch (ObsException e) {
            System.out.println("listVersions failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        }
    }
}
```

```
    } catch (Exception e) {  
        System.out.println("listVersions failed");  
        // 其他异常信息打印  
        e.printStackTrace();  
    }  
}
```

## 代码示例：指定起始位置列举多版本对象

以下代码展示如何指定起始位置列举examplebucket桶中对象名字典序在test之后的100个多版本对象。

```
import com.obs.services.ObsClient;  
import com.obs.services.exception.ObsException;  
import com.obs.services.model.ListVersionsRequest;  
import com.obs.services.model.ListVersionsResult;  
import com.obs.services.model.VersionOrDeleteMarker;  
public class ListVersions004 {  
    public static void main(String[] args) {  
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。  
        // 您可以登录访问管理控制台获取访问密钥AK/SK  
        String ak = System.getenv("ACCESS_KEY_ID");  
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");  
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。  
        // String securityToken = System.getenv("SECURITY_TOKEN");  
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。  
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";  
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。  
        //String endPoint = System.getenv("ENDPOINT");  
  
        // 创建ObsClient实例  
        // 使用永久AK/SK初始化客户端  
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);  
        // 使用临时AK/SK和SecurityToken初始化客户端  
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);  
  
        try {  
            // 指定起始位置列举  
            // 列举对象名字典序在"test"之后的100个多版本对象  
            ListVersionsRequest request = new ListVersionsRequest("examplebucket", 100);  
            request.setKeyMarker("test");  
            ListVersionsResult result = obsClient.listVersions(request);  
            System.out.println("listVersions successfully");  
            for (VersionOrDeleteMarker v : result.getVersions()) {  
                System.out.println("Key:" + v.getKey());  
                System.out.println("Owner:" + v.getOwner());  
                System.out.println("isDeleteMarker:" + v.isDeleteMarker());  
            }  
        } catch (ObsException e) {  
            System.out.println("listVersions failed");  
            // 请求失败,打印http状态码  
            System.out.println("HTTP Code:" + e.getResponseCode());  
            // 请求失败,打印服务端错误码  
            System.out.println("Error Code:" + e.getErrorCode());  
            // 请求失败,打印详细错误信息  
            System.out.println("Error Message:" + e.getErrorMessage());  
            // 请求失败,打印请求id  
            System.out.println("Request ID:" + e.getErrorRequestId());  
            System.out.println("Host ID:" + e.getErrorHostId());  
            e.printStackTrace();  
        } catch (Exception e) {  
            System.out.println("listVersions failed");  
            // 其他异常信息打印  
            e.printStackTrace();  
        }  
    }  
}
```



```
}  
}
```

## 代码示例：分页列举全部多版本对象

以下代码展示如何指定起始位置列举examplebucket桶中对象名字典序在test之后的100个多版本对象。

```
import com.obs.services.ObsClient;  
import com.obs.services.exception.ObsException;  
import com.obs.services.model.ListVersionsRequest;  
import com.obs.services.model.ListVersionsResult;  
import com.obs.services.model.VersionOrDeleteMarker;  
public class ListVersions005 {  
    public static void main(String[] args) {  
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。  
        // 您可以登录访问管理控制台获取访问密钥AK/SK  
        String ak = System.getenv("ACCESS_KEY_ID");  
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");  
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。  
        // String securityToken = System.getenv("SECURITY_TOKEN");  
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。  
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";  
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。  
        //String endPoint = System.getenv("ENDPOINT");  
  
        // 创建ObsClient实例  
        // 使用永久AK/SK初始化客户端  
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);  
        // 使用临时AK/SK和SecurityToken初始化客户端  
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);  
  
        try {  
            // 分页列举全部对象  
            ListVersionsResult result;  
            ListVersionsRequest request = new ListVersionsRequest("examplebucket", 100);  
            do {  
                result = obsClient.listVersions(request);  
                System.out.println("listVersions successfully");  
                for (VersionOrDeleteMarker v : result.getVersions()) {  
                    System.out.println("Key:" + v.getKey());  
                    System.out.println("Owner:" + v.getOwner());  
                    System.out.println("isDeleteMarker:" + v.isDeleteMarker());  
                }  
                request.setKeyMarker(result.getNextKeyMarker());  
                request.setVersionIdMarker(result.getNextVersionIdMarker());  
            } while (result.isTruncated());  
        } catch (ObsException e) {  
            System.out.println("listVersions failed");  
            // 请求失败,打印http状态码  
            System.out.println("HTTP Code:" + e.getResponseCode());  
            // 请求失败,打印服务端错误码  
            System.out.println("Error Code:" + e.getErrorCode());  
            // 请求失败,打印详细错误信息  
            System.out.println("Error Message:" + e.getErrorMessage());  
            // 请求失败,打印请求id  
            System.out.println("Request ID:" + e.getErrorRequestId());  
            System.out.println("Host ID:" + e.getErrorHostId());  
            e.printStackTrace();  
        } catch (Exception e) {  
            System.out.println("listVersions failed");  
            // 其他异常信息打印  
            e.printStackTrace();  
        }  
    }  
}
```

## 代码示例：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多版本对象：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListVersionsRequest;
import com.obs.services.model.ListVersionsResult;
import com.obs.services.model.VersionOrDeleteMarker;
public class ListVersions006 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 列举文件夹中的所有对象
            ListVersionsResult result;
            ListVersionsRequest request = new ListVersionsRequest("examplebucket", 100);
            // 设置文件夹对象名"dir/"为前缀
            request.setPrefix("dir/");
            do {
                result = obsClient.listVersions(request);
                System.out.println("listVersions successfully");
                for (VersionOrDeleteMarker v : result.getVersions()) {
                    System.out.println("Key:" + v.getKey());
                    System.out.println("Owner:" + v.getOwner());
                    System.out.println("isDeleteMarker:" + v.isDeleteMarker());
                }
                request.setKeyMarker(result.getNextKeyMarker());
                request.setVersionIdMarker(result.getNextVersionIdMarker());
            } while (result.isTruncated());
        } catch (ObsException e) {
            System.out.println("listVersions failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("listVersions failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 代码示例：按文件夹分组列举所有多版本对象

以下代码展示如何按文件夹分组，列举examplebucket桶下的所有多版本对象：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ListVersionsRequest;
import com.obs.services.model.ListVersionsResult;
import com.obs.services.model.VersionOrDeleteMarker;
public class ListVersions007 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 按文件夹分组列举所有对象
            ListVersionsRequest request = new ListVersionsRequest("examplebucket", 1000);
            request.setDelimiter("/");
            ListVersionsResult result = obsClient.listVersions(request);
            System.out.println("listVersions successfully");
            System.out.println("Objects in the root directory:");
            for (VersionOrDeleteMarker v : result.getVersions()) {
                System.out.println("Key:" + v.getKey());
                System.out.println("Owner:" + v.getOwner());
                System.out.println("isDeleteMarker:" + v.isDeleteMarker());
            }
            listVersionsByPrefix(obsClient, result);
        } catch (ObsException e) {
            System.out.println("listVersions failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("listVersions failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 代码示例：递归列出子文件夹中多版本对象

递归列出子文件夹中多版本对象的函数 *listVersionsByPrefix* 的示例代码如下：

```
// 递归列出子文件夹中多版本对象
static void listVersionsByPrefix(ObsClient obsClient, ListVersionsResult result) throws ObsException {
```

```
for (String prefix : result.getCommonPrefixes()) {
    System.out.println("Objects in folder [" + prefix + "]:");
    ListVersionsRequest request = new ListVersionsRequest("examplebucket", 1000);
    request.setDelimiter("/");
    request.setPrefix(prefix);
    result = obsClient.listVersions(request);
    for (VersionOrDeleteMarker v : result.getVersions()) {
        System.out.println("Key:" + v.getKey());
        System.out.println("Owner:" + v.getOwner());
        System.out.println("isDeleteMarker:" + v.isDeleteMarker());
    }
    listVersionsByPrefix(obsClient, result);
}
}
```

#### 📖 说明

- 以上代码示例不包含文件夹中多版本对象数超过1000个的情况。
- 由于是需要列举出文件夹中的对象和子文件夹，且文件夹对象总是以“/”结尾，因此 delimiter总是为“/”。
- 每次递归的返回结果中ListVersionsResult.getVersions包含的是文件夹中的多版本对象；ListVersionsResult.getCommonPrefixes包含的是文件夹的子文件夹。

## 相关链接

- 关于列举对象的其他说明，请参见[授权码列举对象](#)。
- 更多关于列举多版本对象的示例代码，请参见[Github示例](#)。
- 获取桶列表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](#)。

## 13.8 设置多版本对象 ACL(Java SDK)

### 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)。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及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100条Grant授权。

## 方法定义

obsClient.setObjectAcl([SetObjectAclRequest request](#))

## 请求参数说明

表 13-51 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">SetObjectAclRequest</a>	必选	<b>参数解释：</b> 设置对象ACL请求参数，详见 <a href="#">SetObjectAclRequest</a> 。

表 13-52 SetObjectAclRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释：</b> 桶名。 <b>约束限制：</b> <ul style="list-style-type: none"><li>• 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。</li><li>• 桶命名规则如下：<ul style="list-style-type: none"><li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>- 禁止使用IP地址。</li><li>- 禁止以“-”或“.”开头及结尾。</li><li>- 禁止两个“.”相邻（如：“my..bucket”）。</li><li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li></ul></li><li>• 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。</li></ul> <b>默认取值：</b> 无

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 指定对象的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据 <a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>

表 13-53 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	<a href="#">Owner</a>	可选	<p><b>参数解释:</b> 桶所有者的信息，详见<a href="#">Owner</a>。</p>

参数名称	参数类型	是否必选	参数类型
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。 <b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。 <b>默认取值:</b> false
grants	Set< <a href="#">GrantAndPermission</a> >	可选	<b>参数解释:</b> 被授权用户相关信息，详见 <a href="#">GrantAndPermission</a> 。

表 13-54 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID，即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 13-55 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组，详见 <a href="#">GranteeInterface</a> 。

参数名称	参数类型	是否必选	描述
permission	<a href="#">Permission</a>	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 13-56 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<p><b>参数解释:</b> 被授权用户的用户信息，详见<a href="#">CanonicalGrantee</a>。</p>
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<p><b>参数解释:</b> 被授权用户组的用户组信息。</p> <p><b>取值范围:</b> 详见<a href="#">GroupGrantee</a>。</p> <p><b>默认取值:</b> 无</p>



表 13-57 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<p><b>参数解释:</b> 被授权用户的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
displayName	String	可选	<p><b>参数描述:</b> 被授权用户的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 13-58 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 13-59 Permission

常量名	原始值	说明
PERMISSION_READ	READ	<p>读权限。</p> <p>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。</p> <p>如果有对象的读权限, 则可以获取该对象内容和元数据。</p>
PERMISSION_WRITE	WRITE	<p>写权限。</p> <p>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。</p> <p>此权限在对象上不适用。</p>

常量名	原始值	说明
PERMISSION_READ_ACP	READ_ACP	<p>读取ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有读对应桶或对象ACP的权限。</p>
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

表 13-60 ACL 预定义访问策略

常量名	说明
AccessControlList.REST_CANNED_PRIVATE	<p>私有读写。</p> <p>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。</p>
AccessControlList.REST_CANNED_PUBLIC_READ	<p>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。</p> <p>设在对象上，所有人可以获取该对象内容和元数据。</p>

常量名	说明
AccessControllist.REST_CAN NED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControllist.REST_CAN NED_PUBLIC_READ_DELIVER ED	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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControllist.REST_CAN NED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

## 返回结果说明

表 13-61 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHead ers	Map<String, Object>	<b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无

## 代码示例

本示例用于为examplebucket桶中objectname对象设置多版本对象访问权限。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
import com.obs.services.model.GroupGrantee;
import com.obs.services.model.Owner;
import com.obs.services.model.Permission;
public class SetObjectAcl001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 通过预定义访问策略设置多版本对象访问权限为私有读写
            obsClient.setObjectAcl("examplebucket", "objectname", AccessControlList.REST_CANNED_PRIVATE, "versionid");
            AccessControlList acl = new AccessControlList();
            Owner owner = new Owner();
            owner.setId("ownerid");
            acl.setOwner(owner);
            // 为所有用户设置读权限
            acl.grantPermission(GroupGrantee.ALL_USERS, Permission.PERMISSION_READ);
            // 设置多版本对象访问权限
            obsClient.setObjectAcl("examplebucket", "objectname", acl, "versionid");
            System.out.println("setObjectAcl successfully");
        } catch (ObsException e) {
            System.out.println("setObjectAcl failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("setObjectAcl failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

### 📖 说明

ACL中需要填写的所有者（Owner）或者被授权用户（Grantee）的ID，是指用户的账号ID，可通过OBS控制台“我的凭证”页面查看。

## 相关链接

- 关于设置对象ACL的API说明，请参见[设置对象ACL](#)。
- 更多关于设置对象ACL的示例代码，请参见[Github示例](#)。
- 设置对象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](#)。

## 13.9 获取多版本对象权限(Java SDK)

### 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。OBS用户可以通过ACL操作API接口对已存在的对象更改或者获取ACL(access control list)。

调用获取桶的获取多版本对象权限接口，将获取指定桶的获取多版本对象权限。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 用户必须拥有对指定对象读ACP(access control policy)的权限，才能执行获取对象ACL的操作。

### 方法定义

obsClient.getObjectAcl([GetObjectAclRequest request](#))

### 请求参数说明

表 13-62 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">GetObjectAclRequest</a>	必选	参数解释： 获取对象ACL请求参数，详见 <a href="#">GetObjectAclRequest</a> 。

表 13-63 GetObjectAclRequest 请求参数列表

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	必选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 13-64 AccessControlList

参数名称	参数类型	是否必选	描述
owner	<b>Owner</b>	可选	<b>参数解释:</b> 桶所有者的信息, 详见 <b>Owner</b> 。
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false
grants	Set< <b>GrantAndPermission</b> >	可选	<b>参数解释:</b> 被授权用户相关信息, 详见 <b>GrantAndPermission</b> 。

表 13-65 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> <b>默认取值:</b> 无

表 13-66 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组, 详见 <a href="#">GranteeInterface</a> 。
permission	<a href="#">Permission</a>	必选	<b>参数解释:</b> 用户或用户组被授予的权限。 <b>取值范围:</b> 详见 <a href="#">Permission</a> 。 <b>默认取值:</b> 无
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false

表 13-67 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无



表 13-68 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<p><b>参数解释:</b> 被授权用户的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
displayName	String	可选	<p><b>参数描述:</b> 被授权用户的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 13-69 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 13-70 Permission

常量名	原始值	说明
PERMISSION_READ	READ	<p>读权限。</p> <p>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。</p> <p>如果有对象的读权限, 则可以获取该对象内容和元数据。</p>
PERMISSION_WRITE	WRITE	<p>写权限。</p> <p>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。</p> <p>此权限在对象上不适用。</p>

常量名	原始值	说明
PERMISSION_READ_ACP	READ_ACP	<p>读取ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有读对应桶或对象ACP的权限。</p>
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

## 代码示例

获取examplebucket桶内objectname对象多版本对象访问权限。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
public class GetObjectAcl001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
```

```
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 获取多版本对象访问权限
    AccessControlList acl = obsClient.getObjectAcl("examplebucket", "objectname", "versionid");
    System.out.println("getObjectAcl successfully");
    System.out.println(acl);
} catch (ObsException e) {
    System.out.println("getObjectAcl failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getObjectAcl failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于获取对象ACL的API说明，请参见[获取对象ACL](#)。
- 更多关于获取对象ACL的示例代码，请参见[Github示例](#)。
- 获取对象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](#)。

## 13.10 删除多版本对象(Java SDK)

### 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对象。

您可以通过ObsClient.deleteObject接口传入版本号（versionId）删除多版本对象。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及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([DeleteObjectRequest request](#))

## 请求参数说明

表 13-71 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">DeleteObjectRequest</a>	必选	<b>参数解释：</b> 删除对象请求参数，详见 <a href="#">DeleteObjectRequest</a> 。

表 13-72 DeleteObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值：</b> 无</p>

参数名称	参数类型	是否必选	描述
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 13-73 DeleteObjectResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
versionId	String	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>
deleteMarker	boolean	<p><b>参数解释:</b> 标识删除的对象是否是删除标记。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• true: 是删除标记。</li> <li>• false: 不是删除标记。</li> </ul> <p><b>默认取值:</b> false</p>
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

您可以通过ObsClient.deleteObject接口传入版本号（versionId）删除examplebucket桶中objectname的多版本对象，示例代码如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class DeleteObject001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
```

```
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk,endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 删除单个对象
    obsClient.deleteObject("examplebucket", "objectname", "versionid");
    System.out.println("deleteObject successfully");
} catch (ObsException e) {
    System.out.println("deleteObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("deleteObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于删除对象的API说明，请参见[删除对象](#)。
- 更多关于删除对象的示例代码，请参见[Github示例](#)。
- 删除对象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](#)。

## 13.11 批量删除多版本对象(Java SDK)

### 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。

您可以通过ObsClient.deleteObjects接口传入版本号（versionId）删除多版本对象。

### 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以及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([DeleteObjectsRequest deleteRequest](#))

## 请求参数说明

表 13-74 请求参数列表

参数名称	参数类型	是否必选	描述
deleteRequest	<a href="#">DeleteObjectsRequest</a>	必选	<b>参数解释：</b> 批量删除对象请求参数，详见 <a href="#">DeleteObjectsRequest</a> 。

表 13-75 DeleteObjectsRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释：</b> 桶名。 <b>约束限制：</b> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 桶命名规则如下： <ul style="list-style-type: none"> <li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>- 禁止使用IP地址。</li> <li>- 禁止以“-”或“.”开头及结尾。</li> <li>- 禁止两个“.”相邻（如：“my..bucket”）。</li> <li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>• 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。</li> </ul> <b>默认取值：</b> 无



参数名称	参数类型	是否必选	描述
keyAndVersions	List<KeyAndVersion>	必选	<b>参数解释:</b> 待删除的对象列表, 详情参考 <a href="#">KeyAndVersion</a> 。
quiet	boolean	可选	<b>参数解释:</b> 批量删除对象的响应方式。 <b>取值范围:</b> <ul style="list-style-type: none"> <li>• false: 表示详细模式, 返回删除成功和删除失败的所有结果。</li> <li>• true: 表示简单模式, 只返回删除过程中出错的结果。</li> </ul> <b>默认取值:</b> false
encodingType	String	可选	<b>参数解释:</b> 用于指定对响应中的对象名objectKey进行指定类型的编码。如果objectKey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objectKey进行编码。 <b>取值范围:</b> 可选值为url。 <b>默认取值:</b> 无, 不设置则不编码。

表 13-76 KeyAndVersion

参数名称	参数类型	是否必选	描述
key	String	必选	<b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中, 对象名为folder/test.txt。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号，用于删除指定版本号的对象。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无，如果不设置则默认删除最新版本的对象。</p>

## 返回结果说明

表 13-77 DeleteObjectsResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>
deletedObjectResults	List< <a href="#">DeleteObjectResult</a> >	<p><b>参数解释:</b> 删除对象后的响应结果，详见<a href="#">DeleteObjectResult</a>。</p>
errorResults	List< <a href="#">ErrorResult</a> >	<p><b>参数解释:</b> 删除失败的对象列表，详见<a href="#">ErrorResult</a>。</p>

表 13-78 DeleteObjectResult

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无
versionId	String	<b>参数解释:</b> 对象的版本号。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
deleteMarker	boolean	<b>参数解释:</b> 标识删除的对象是否是删除标记。 <b>取值范围:</b> <ul style="list-style-type: none"><li>• true: 是删除标记。</li><li>• false: 不是删除标记。</li></ul> <b>默认取值:</b> false

参数名称	参数类型	描述
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

表 13-79 ErrorResult

参数名称	参数类型	描述
versionId	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
errorCode	String	<p><b>参数解释:</b> 删除失败的错误码。可参考<a href="#">OBS错误码</a>。</p>

参数名称	参数类型	描述
objectKey	String	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
message	String	<p><b>参数解释:</b> 删除失败的错误原因。可参考<a href="#">OBS错误码</a>。</p>

## 代码示例

您可以通过ObsClient.deleteObjects接口传入每个待删除对象的版本号（versionId）为examplebucket桶批量删除多版本对象，示例代码如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.DeleteObjectsRequest;
import com.obs.services.model.DeleteObjectsResult;
import com.obs.services.model.KeyAndVersion;
import java.util.ArrayList;
import java.util.List;
import java.util.List;
public class DeleteObjects001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 批量删除多版本对象
            DeleteObjectsRequest request = new DeleteObjectsRequest("examplebucket");
            request.setQuiet(false);
            List<KeyAndVersion> toDelete = new ArrayList<KeyAndVersion>();
            toDelete.add(new KeyAndVersion("objectname1", "versionid1"));
        }
    }
}
```

```
toDelete.add(new KeyAndVersion("objectname2", "versionid2"));
toDelete.add(new KeyAndVersion("objectname3", "versionid3"));
request.setKeyAndVersions(toDelete.toArray(new KeyAndVersion[toDelete.size()]));
DeleteObjectsResult result = obsClient.deleteObjects(request);
System.out.println("deleteObjects successfully");
System.out.println("getDeletedObjectResults:" + result.getDeletedObjectResults());
System.out.println("getErrorResults:" + result.getErrorResults());
} catch (ObsException e) {
    System.out.println("deleteObjects failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("deleteObjects failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于批量删除对象的API说明, 请参见[批量删除对象](#)。
- 更多关于删除对象的示例代码, 请参见[Github示例](#)。
- 删除对象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](#)。

# 14 生命周期管理(Java SDK)

## 14.1 生命周期管理简介(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

OBS允许您对桶设置生命周期规则，实现自动转换对象的存储类别、自动淘汰过期的对象，以有效利用存储特性，优化存储空间。针对不同前缀的对象，您可以同时设置多条规则。一条规则包含：

- 规则ID，用于标识一条规则，不能重复。
- 受影响的对象前缀，此规则只作用于符合前缀的对象。
- 最新版本对象的转换策略，指定方式为：
  - a. 指定满足前缀的对象创建后第几天时转换为指定的存储类别。
  - b. 直接指定满足前缀的对象转换为指定的存储类别的日期。
- 最新版本对象过期时间，指定方式为：
  - a. 指定满足前缀的对象创建后第几天时过期。
  - b. 直接指定满足前缀的对象过期日期。
- 历史版本对象转换策略，指定方式为：
  - 指定满足前缀的对象成为历史版本后第几天时转换为指定的存储类别。
- 历史版本对象过期时间，指定方式为：
  - 指定满足前缀的对象成为历史版本后第几天时过期。
- 是否生效标识。

更多关于生命周期的内容请参考[生命周期管理](#)。

## 14.2 设置生命周期规则(Java SDK)

### 功能介绍

OBS支持用户调用设置桶的生命周期配置接口，为指定桶设置生命周期规则，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](#)。

1. 对象过期后会被OBS服务端自动删除。
2. 对象转换策略中的时间必须早于对象过期时间；历史版本对象转换策略中的时间也必须早于历史版本对象的过期时间。
3. 桶的多版本状态必须处于Enabled或者Suspended，历史版本对象转换策略和历史版本对象过期时间配置才能生效。

#### 说明

- 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。如果对象经过转换，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。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- 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。

### 方法定义

obsClient.setBucketLifecycle(final [SetBucketLifecycleRequest request](#))



## 请求参数说明

表 14-1 请求参数列表

参数名称	参数类型	是否必选	描述
deleteRequest	<a href="#">SetBucketLifecycleRequest</a>	必选	<b>参数解释：</b> 设置生命周期请求参数，详见 <a href="#">SetBucketLifecycleRequest</a> 。

表 14-2 SetBucketLifecycleRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释：</b> 桶名。 <b>约束限制：</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li></ul></li><li>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。</li></ul> <b>默认取值：</b> 无
lifecycleConfiguration	<a href="#">LifecycleConfiguration</a>	必选	<b>参数解释：</b> 桶生命周期规则列表，详见 <a href="#">LifecycleConfiguration</a> 。

表 14-3 LifecycleConfiguration

参数名称	参数类型	是否必选	描述
rules	List<Rule>	可选	<b>参数解释:</b> 桶生命周期规则列表, 详见Rule。

表 14-4 Rule

参数名称	参数类型	是否必选	描述
id	String	可选	<b>参数解释:</b> 生命周期规则ID。 <b>取值范围:</b> 长度大于0且不超过255的字符串。 <b>默认取值:</b> 无
prefix	String	必选	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无
enabled	boolean	必选	<b>参数解释:</b> 标识当前规则是否启用。 <b>取值范围:</b> true: 当前规则启用。 false: 当前规则未启用。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
expiration	<a href="#">Expiration</a>	可选	<p><b>参数解释:</b> 对象过期时间配置。详见<a href="#">Expiration</a>。</p> <p><b>默认取值:</b> 无</p>
noncurrentVersionExpiration	<a href="#">NoncurrentVersionExpiration</a>	可选	<p><b>参数解释:</b> 历史版本对象过期时间配置，详见<a href="#">NoncurrentVersionExpiration</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 仅针对对象的历史版本。</li> <li>• 当前桶已启用（或开启后暂停）多版本。</li> </ul> <p><b>默认取值:</b> 无</p>
transitions	List< <a href="#">Transition</a> >	可选	<p><b>参数解释:</b> 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<a href="#">Transition</a>。</p> <p><b>约束限制:</b> 仅针对对象的最新版本。</p> <p><b>默认取值:</b> 无</p>
noncurrentVersionTransitions	List< <a href="#">NoncurrentVersionTransition</a> >	可选	<p><b>参数解释:</b> 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<a href="#">NoncurrentVersionTransition</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 仅针对对象的历史版本。</li> <li>• 当前桶已启用（或开启后暂停）多版本。</li> </ul> <p><b>默认取值:</b> 无</p>

表 14-5 Expiration

参数名称	参数类型	是否必选	描述
days	Integer	如果没有设置 date 则必选	<b>参数解释:</b> 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。 <b>约束限制:</b> 仅适用于对象的最新版本。 <b>取值范围:</b> 正整数，单位：天。 <b>默认取值:</b> 无
date	Date	如果没有设置 days 则必选	<b>参数解释:</b> 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。 <b>默认取值:</b> 无

表 14-6 NoncurrentVersionExpiration

参数名称	参数类型	是否必选	描述
days	Integer	必选	<b>参数解释:</b> 表示对象成为历史版本后第几天时过期。 <b>约束限制:</b> 仅针对历史版本。 <b>取值范围:</b> 正整数，单位：天。 <b>默认取值:</b> 无

表 14-7 Transition

参数名称	参数类型	是否必选	描述
days	Integer	如果没有设置date则必选	<p><b>参数解释:</b> 表示在对象创建时间后第几天时转换。</p> <p><b>约束限制:</b> 仅适用于对象的最新版本。</p> <p><b>取值范围:</b> 正整数, 单位: 天。</p> <p><b>默认取值:</b> 无</p>
date	Date	如果没有设置days则必选	<p><b>参数解释:</b> 表示对象转换的日期。</p> <p><b>默认取值:</b> 无</p>
storageClass	<b>StorageClassEnum</b>	必选	<p><b>参数解释:</b> 对象转换后的存储类别。</p> <p><b>约束限制:</b> 对象存储类别转换限制:</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>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。</li> </ul> <p><b>取值范围:</b> 生命周期转换的目标存储类别取值范围详见 <b>StorageClassEnum</b>。</p> <p><b>默认取值:</b> 无</p>

表 14-8 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。

常量名	原始值	说明
COLD	COLD	归档存储。

表 14-9 NoncurrentVersionTransition

参数名称	参数类型	是否必选	描述
days	Integer	如果没有设置date则必选	<p><b>参数解释:</b> 表示在对象创建时间后第几天时转换。</p> <p><b>约束限制:</b> 仅适用于历史版本。</p> <p><b>取值范围:</b> 正整数，单位：天。</p> <p><b>默认取值:</b> 无</p>
storageClass	<a href="#">StorageClassEnum</a>	必选	<p><b>参数解释:</b> 对象转换后的存储类别。</p> <p><b>约束限制:</b> 对象存储类别转换限制：</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>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。</li> </ul> <p><b>取值范围:</b> 生命周期转换的目标存储类别取值范围详见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>

#### 📖 说明

Transitions、Expiration、NoncurrentVersionTransitions、NoncurrentVersionExpiration、AbortIncompleteMultipartUpload不能全为空。

## 返回结果说明

表 14-10 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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 <a href="#">状态码</a> 。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。 <b>默认取值:</b> 无

### 代码示例：设置对象转换策略

以下代码展示了如何设置examplebucket桶最新版本对象和历史版本对象的转换策略。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.LifecycleConfiguration;
import com.obs.services.model.StorageClassEnum;
public class SetBucketLifecycle001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 设置对象转换策略
```

```
LifecycleConfiguration config = new LifecycleConfiguration();
LifecycleConfiguration.Rule rule = config.new Rule();
rule.setEnabled(true);
rule.setId("rule1");
rule.setPrefix("prefix");
LifecycleConfiguration.Transition transition = config.new Transition();
// 指定满足前缀的对象创建30天后转换
transition.setDays(30);
// 指定对象转换后的存储类型
transition.setObjectStorageClass(StorageClassEnum.WARM);
// 直接指定满足前缀的对象转换日期
// transition.setDate(new SimpleDateFormat("yyyy-MM-dd").parse("2018-10-31"));
rule.getTransitions().add(transition);
LifecycleConfiguration.NoncurrentVersionTransition noncurrentVersionTransition =
    config.new NoncurrentVersionTransition();
// 指定满足前缀的对象成为历史版本30天后转换
noncurrentVersionTransition.setDays(30);
// 指定历史版本对象转换后的存储类型
noncurrentVersionTransition.setObjectStorageClass(StorageClassEnum.COLD);
rule.getNoncurrentVersionTransitions().add(noncurrentVersionTransition);
// 设置碎片过期时间
LifecycleConfiguration.AbortIncompleteMultipartUpload abortIncompleteMultipartUpload =
config.new AbortIncompleteMultipartUpload();
abortIncompleteMultipartUpload.setDaysAfterInitiation(7);
rule.setAbortIncompleteMultipartUpload(abortIncompleteMultipartUpload);
config.addRule(rule);
obsClient.setBucketLifecycle("examplebucket", config);
System.out.println("setBucketLifecycle successfully");
} catch (ObsException e) {
    System.out.println("setBucketLifecycle failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("setBucketLifecycle failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：设置对象过期时间

以下代码展示了如何设置examplebucket桶最新版本对象和历史版本对象的过期时间。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.LifecycleConfiguration;
public class SetBucketLifecycle002 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
```



```
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk,endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 设置对象过期时间
    LifecycleConfiguration config = new LifecycleConfiguration();
    LifecycleConfiguration.Rule rule = config.new Rule();
    rule.setEnabled(true);
    rule.setId("rule1");
    rule.setPrefix("prefix");
    LifecycleConfiguration.Expiration expiration = config.new Expiration();
    // 指定满足前缀的对象创建60天后过期
    expiration.setDays(60);
    // 直接指定满足前缀的对象过期时间
    // expiration.setDate(new SimpleDateFormat("yyyy-MM-dd").parse("2018-12-31"));
    rule.setExpiration(expiration);
    LifecycleConfiguration.NoncurrentVersionExpiration noncurrentVersionExpiration =
        config.new NoncurrentVersionExpiration();
    // 指定满足前缀的对象成为历史版本60天后过期
    noncurrentVersionExpiration.setDays(60);
    rule.setNoncurrentVersionExpiration(noncurrentVersionExpiration);
    config.addRule(rule);
    obsClient.setBucketLifecycle("examplebucket", config);
    System.out.println("setBucketLifecycle successfully");
} catch (ObsException e) {
    System.out.println("setBucketLifecycle failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("setBucketLifecycle failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置桶的生命周期配置的API说明，请参见[设置桶的生命周期配置](#)。
- 更多关于设置桶的生命周期配置的示例代码，请参见[Github示例](#)。
- 设置桶的生命周期配置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](#)。
- 生命周期管理相关问题请参见[生命周期管理相关问题](#)。

## 14.3 获取生命周期规则(Java SDK)

### 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。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及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(final [BaseBucketRequest request](#))

### 请求参数说明

表 14-11 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">BaseBucketRequest</a>	必选	参数解释： 桶的基本信息请求参数列表，详见 <a href="#">BaseBucketRequest</a> 。

表 14-12 BaseBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 14-13 LifecycleConfiguration

参数名称	参数类型	描述
rules	List< <a href="#">Rule</a> >	<p><b>参数解释:</b> 桶生命周期规则列表，详见<a href="#">Rule</a>。</p>

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

表 14-14 Rule

参数名称	参数类型	是否必选	描述
id	String	可选	<p><b>参数解释:</b> 生命周期规则ID。</p> <p><b>取值范围:</b> 长度大于0且不超过255的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
prefix	String	必选	<p><b>参数解释:</b> 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><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
enabled	boolean	必选	<p><b>参数解释:</b> 标识当前规则是否启用。</p> <p><b>取值范围:</b> true: 当前规则启用。 false: 当前规则未启用。</p> <p><b>默认取值:</b> 无</p>
expiration	<a href="#">Expiration</a>	可选	<p><b>参数解释:</b> 对象过期时间配置。详见<a href="#">Expiration</a>。</p> <p><b>默认取值:</b> 无</p>
noncurrentVersionExpiration	<a href="#">NoncurrentVersionExpiration</a>	可选	<p><b>参数解释:</b> 历史版本对象过期时间配置,详见<a href="#">NoncurrentVersionExpiration</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>仅针对对象的历史版本。</li> <li>当前桶已启用(或开启后暂停)多版本。</li> </ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
transitions	List<Transition>	可选	<p><b>参数解释:</b> 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<a href="#">Transition</a>。</p> <p><b>约束限制:</b> 仅针对对象的最新版本。</p> <p><b>默认取值:</b> 无</p>
noncurrentVersionTransitions	List<NoncurrentVersionTransition>	可选	<p><b>参数解释:</b> 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<a href="#">NoncurrentVersionTransition</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>仅针对对象的历史版本。</li> <li>当前桶已启用（或开启后暂停）多版本。</li> </ul> <p><b>默认取值:</b> 无</p>

表 14-15 Expiration

参数名称	参数类型	是否必选	描述
days	Integer	如果没有设置 date 则必选	<p><b>参数解释:</b> 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><b>约束限制:</b> 仅适用于对象的最新版本。</p> <p><b>取值范围:</b> 正整数，单位：天。</p> <p><b>默认取值:</b> 无</p>
date	Date	如果没有设置 days 则必选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>

表 14-16 NoncurrentVersionExpiration

参数名称	参数类型	是否必选	描述
days	Integer	必选	<p><b>参数解释:</b> 表示对象成为历史版本后第几天时过期。</p> <p><b>约束限制:</b> 仅针对历史版本。</p> <p><b>取值范围:</b> 正整数，单位：天。</p> <p><b>默认取值:</b> 无</p>

表 14-17 Transition

参数名称	参数类型	是否必选	描述
days	Integer	如果没有设置date则必选	<p><b>参数解释:</b> 表示在对象创建时间后第几天时转换。</p> <p><b>约束限制:</b> 仅适用于对象的最新版本。</p> <p><b>取值范围:</b> 正整数，单位：天。</p> <p><b>默认取值:</b> 无</p>
date	Date	如果没有设置days则必选	<p><b>参数解释:</b> 表示对象转换的日期。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
storageClass	<a href="#">StorageClassEnum</a>	必选	<p><b>参数解释：</b> 对象转换后的存储类别。</p> <p><b>约束限制：</b> 对象存储类别转换限制：</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>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。</li> </ul> <p><b>取值范围：</b> 生命周期转换的目标存储类别取值范围详见<a href="#">StorageClassEnum</a>。</p> <p><b>默认取值：</b> 无</p>

表 14-18 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。



表 14-19 NoncurrentVersionTransition

参数名称	参数类型	是否必选	描述
days	Integer	如果没有设置date则必选	<p><b>参数解释:</b> 表示在对象创建时间后第几天时转换。</p> <p><b>约束限制:</b> 仅适用于历史版本。</p> <p><b>取值范围:</b> 正整数，单位：天。</p> <p><b>默认取值:</b> 无</p>
storageClass	<b>StorageClassEnum</b>	必选	<p><b>参数解释:</b> 对象转换后的存储类别。</p> <p><b>约束限制:</b> 对象存储类别转换限制：</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>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。</li> </ul> <p><b>取值范围:</b> 生命周期转换的目标存储类别取值范围详见 <a href="#">StorageClassEnum</a>。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

以下代码展示了如何获取examplebucket桶的生命周期规则。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.LifecycleConfiguration;
public class GetBucketLifecycle001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
// String securityToken = System.getenv("SECURITY_TOKEN");
// endpoint填写桶所在的endpoint, 此处以华北-北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk,endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 查看生命周期规则
    LifecycleConfiguration config = obsClient.getBucketLifecycle("examplebucket");
    System.out.println("getBucketLifecycle successfully");
    for (LifecycleConfiguration.Rule rule : config.getRules()) {
        System.out.println(rule.getId());
        System.out.println(rule.getPrefix());
        for (LifecycleConfiguration.Transition transition : rule.getTransitions()) {
            System.out.println(transition.getDays());
            System.out.println(transition.getStorageClass());
        }
        System.out.println(rule.getExpiration() != null ? rule.getExpiration().getDays() : "");
        for (LifecycleConfiguration.NoncurrentVersionTransition noncurrentVersionTransition :
            rule.getNoncurrentVersionTransitions()) {
            System.out.println(noncurrentVersionTransition.getDays());
            System.out.println(noncurrentVersionTransition.getStorageClass());
        }
        System.out.println(
            rule.getNoncurrentVersionExpiration() != null
                ? rule.getNoncurrentVersionExpiration().getDays()
                : "");
    }
} catch (ObsException e) {
    System.out.println("getBucketLifecycle failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getBucketLifecycle failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于获取桶的生命周期配置的API说明，请参见[获取桶的生命周期配置](#)。
- 更多关于获取桶的生命周期配置的示例代码，请参见[Github示例](#)。
- 获取桶的生命周期配置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](#)。
- 生命周期管理相关问题请参见[生命周期管理相关问题](#)。

## 14.4 删除生命周期规则(Java SDK)

### 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。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及region与endPoint的对应关系，详细信息请参见[地区与终端节点](#)。
- Java SDK暂不支持指定规则ID删除，调用删除生命周期规则接口时，将删除桶内所有的生命周期规则。

### 方法定义

obsClient.deleteBucketLifecycle(final [BaseBucketRequest request](#))

### 请求参数说明

表 14-20 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">BaseBucketRequest</a>	必选	参数解释： 桶的基本信息请求参数列表，详见 <a href="#">BaseBucketRequest</a> 。

表 14-21 BaseBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 14-22 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

以下代码展示了如何删除examplebucket桶的生命周期规则。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class DeleteBucketLifecycle001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 删除生命周期规则
            obsClient.deleteBucketLifecycle("examplebucket");
            System.out.println("deleteBucketLifecycle successfully");
        } catch (ObsException e) {
            System.out.println("deleteBucketLifecycle failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("deleteBucketLifecycle failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于删除桶的生命周期配置的API说明，请参见[删除桶的生命周期配置](#)。
- 更多关于删除桶的生命周期配置的示例代码，请参见[Github示例](#)。
- 删除桶的生命周期配置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](#)。
- 生命周期管理相关问题请参见[生命周期管理相关问题](#)。

# 15 跨域资源共享(Java SDK)

## 15.1 跨域资源共享简介(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

跨域资源共享（CORS）允许Web端的应用程序访问不属于本域的资源。OBS提供接口方便开发者控制跨域访问的权限。

更多关于跨域资源共享的内容请参考[跨域资源访问](#)。

## 15.2 设置跨域资源共享规则(Java SDK)

### 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。

您可以通过ObsClient.setBucketCors设置桶的跨域规则，如果原规则存在则覆盖原规则。

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 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以及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(final [SetBucketCorsRequest request](#))

## 请求参数说明

表 15-1 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">SetBucketCorsRequest</a>	必选	设置跨域规则请求参数，详见 <a href="#">SetBucketCorsRequest</a> 。



表 15-2 SetBucketCorsRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值：</b> 无</p>
bucketCors	<a href="#">BucketCors</a>	必选	<p><b>参数解释：</b> 桶的CORS规则列表。详见<a href="#">BucketCors</a>。</p>

表 15-3 BucketCors

参数名称	参数类型	是否必选	描述
rules	List< <a href="#">BucketCorsRule</a> >	必选	<p><b>参数解释：</b> 桶的CORS规则列表。详见<a href="#">BucketCorsRule</a>。</p>

表 15-4 BucketCorsRule

参数名称	参数类型	是否必选	描述
id	String	可选	<b>参数解释:</b> CORS规则ID <b>取值范围:</b> 长度大于0且不超过255的字符串。 <b>默认取值:</b> 无
allowedMethod	List<String>	必选	<b>参数解释:</b> 指定允许的跨域请求HTTP方法，即桶和对象的几种操作类型。 <b>取值范围:</b> 支持以下HTTP方法： <ul style="list-style-type: none"><li>• GET</li><li>• PUT</li><li>• HEAD</li><li>• POST</li><li>• DELETE</li></ul> <b>默认取值:</b> 无
allowedOrigin	List<String>	必选	<b>参数解释:</b> 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。 <b>约束限制:</b> 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。 <b>默认取值:</b> 无
allowedHeader	List<String>	可选	<b>参数解释:</b> 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。 <b>约束限制:</b> 最多可填写一个“*”通配符，不支持&、:、<、空格以及中文字符。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
maxAgeSeconds	int	可选	<p><b>参数解释:</b> CORS规则允许客户端可以对跨域请求返回结果的缓存时间。</p> <p><b>约束限制:</b> 每个BucketCorsRule可以包含至多一个maxAgeSeconds。</p> <p><b>取值范围:</b> 不小于0的整型数，单位：秒。</p> <p><b>默认取值:</b> 100</p>
exposeHeader	List<String>	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 不支持*、&amp;、:、&lt;、空格以及中文字符。</p> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 15-5 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于配置examplebucket桶的CORS规则。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketCors;
import com.obs.services.model.BucketCorsRule;
import com.obs.services.model.DeleteObjectsRequest;
import com.obs.services.model.DeleteObjectsResult;
import com.obs.services.model.KeyAndVersion;
import java.util.ArrayList;
import java.util.List;
public class SetBucketCors001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 设置跨域规则
            BucketCors cors = new BucketCors();
            List<BucketCorsRule> rules = new ArrayList<BucketCorsRule>();
            BucketCorsRule rule = new BucketCorsRule();
            ArrayList<String> allowedOrigin = new ArrayList<String>();
            // 指定允许跨域请求的来源
            allowedOrigin.add( "http://www.a.com");
            allowedOrigin.add( "http://www.b.com");
            rule.setAllowedOrigin(allowedOrigin);
            ArrayList<String> allowedMethod = new ArrayList<String>();
            // 指定允许的跨域请求方法(GET/PUT/DELETE/POST/HEAD)
            allowedMethod.add("GET");
            allowedMethod.add("HEAD");
            allowedMethod.add("PUT");
            rule.setAllowedMethod(allowedMethod);
            ArrayList<String> allowedHeader = new ArrayList<String>();
            // 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.add("x-obs-header");
```

```
rule.setAllowedHeader(allowedHeader);
ArrayList<String> exposeHeader = new ArrayList<String>();
// 指定允许用户从应用程序中访问的header
exposeHeader.add("x-obs-expose-header");
rule.setExposeHeader(exposeHeader);
// 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
rule.setMaxAgeSecond(10);
rules.add(rule);
cors.setRules(rules);
obsClient.setBucketCors("examplebucket", cors);
System.out.println("setBucketCors successfully");
} catch (ObsException e) {
    System.out.println("setBucketCors failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("setBucketCors failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置桶的跨域规则的API说明，请参见[设置桶的CORS配置](#)。
- 更多关于设置桶的跨域规则的示例代码，请参见[Github示例](#)。
- 设置桶的跨域规则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](#)。

## 15.3 获取跨域资源共享规则(Java SDK)

### 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。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 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以及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(final [BaseBucketRequest request](#))

## 请求参数说明

表 15-6 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">BaseBucketRequest</a>	必选	<b>参数解释：</b> 查看跨域规则请求参数，详见 <a href="#">BaseBucketRequest</a> 。

表 15-7 BaseBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 15-8 BucketCors

参数名称	参数类型	是否必选	描述
rules	List<BucketCorsRule>	必选	<p><b>参数解释:</b> 桶的CORS规则列表。详见 <a href="#">BucketCorsRule</a>。</p>

表 15-9 BucketCorsRule

参数名称	参数类型	是否必选	描述
id	String	可选	<p><b>参数解释:</b> CORS规则ID</p> <p><b>取值范围:</b> 长度大于0且不超过255的字符串。</p> <p><b>默认取值:</b> 无</p>
allowedMethod	List<String>	必选	<p><b>参数解释:</b> 指定允许的跨域请求HTTP方法，即桶和对象的几种操作类型。</p> <p><b>取值范围:</b> 支持以下HTTP方法：</p> <ul style="list-style-type: none"> <li>• GET</li> <li>• PUT</li> <li>• HEAD</li> <li>• POST</li> <li>• DELETE</li> </ul> <p><b>默认取值:</b> 无</p>
allowedOrigin	List<String>	必选	<p><b>参数解释:</b> 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><b>约束限制:</b> 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><b>默认取值:</b> 无</p>
allowedHeader	List<String>	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 最多可填写一个“*”通配符，不支持&amp;、:、&lt;、空格以及中文字符。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
maxAgeSeconds	int	可选	<p><b>参数解释:</b> CORS规则允许客户端可以对跨域请求返回结果的缓存时间。</p> <p><b>约束限制:</b> 每个BucketCorsRule可以包含至多一个maxAgeSeconds。</p> <p><b>取值范围:</b> 不小于0的整型数，单位：秒。</p> <p><b>默认取值:</b> 100</p>
exposeHeader	List<String>	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 不支持*、&amp;、:、&lt;、空格以及中文字符。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于获取名为examplebucket桶的CORS规则配置。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketCors;
import com.obs.services.model.BucketCorsRule;
public class GetBucketCors001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);
```

```
try {
    // 获取跨域规则
    BucketCors cors = obsClient.getBucketCors("examplebucket");
    for(BucketCorsRule rule : cors.getRules()){
        System.out.println("Id:" + rule.getId());
        System.out.println("MaxAgeSecond:" + rule.getMaxAgeSecond());
        System.out.println("AllowedHeader:" + rule.getAllowedHeader());
        System.out.println("AllowedOrigin:" + rule.getAllowedOrigin());
        System.out.println("AllowedMethod:" + rule.getAllowedMethod());
        System.out.println("ExposeHeader:" + rule.getExposeHeader());
    }
    System.out.println("getBucketCors successfully");
} catch (ObsException e) {
    System.out.println("getBucketCors failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getBucketCors failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于获取桶的跨域规则的API说明，请参见[获取桶的CORS配置](#)。
- 更多关于获取桶的跨域规则的示例代码，请参见[Github示例](#)。
- 获取桶的跨域规则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](#)。

## 15.4 删除跨域资源共享规则(Java SDK)

### 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。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 接口约束

- 您必须是桶所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以及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(final [BaseBucketRequest request](#))

## 请求参数说明

表 15-10 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">BaseBucketRequest</a>	必选	<b>参数解释：</b> 删除跨域规则请求参数，详见 <a href="#">BaseBucketRequest</a> 。

表 15-11 BaseBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 15-12 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于删除名为examplebucket桶的CORS规则。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class DeleteBucketCors001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 删除跨域规则
            obsClient.deleteBucketCors("examplebucket");
            System.out.println("deleteBucketCors successfully");
        } catch (ObsException e) {
            System.out.println("deleteBucketCors failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("deleteBucketCors failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于删除桶的跨域规则的API说明，请参见[删除桶的CORS配置](#)。
- 更多关于删除桶的跨域规则的示例代码，请参见[Github示例](#)。
- 删除桶的跨域规则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](#)。

# 16 设置访问日志(Java SDK)

## 16.1 日志简介(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

OBS允许您对桶设置访问日志记录，设置之后对于桶的访问会被记录成日志，日志存储在OBS上您指定的目标桶中。

更多关于访问日志的内容请参考[日志记录](#)。

## 16.2 设置桶日志规则(Java SDK)

### 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Java 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 to 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。

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 接口约束

- 日志目标桶（存放日志的桶）与源桶（产生日志的桶）必须在同一个区域（region）。
- 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。
- 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](#)。
- 您必须是桶所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以及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.setBucketLogging(final [SetBucketLoggingRequest request](#))

## 请求参数说明

表 16-1 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">SetBucketLoggingRequest</a>	必选	参数解释： 设置桶日志请求参数，详见 <a href="#">SetBucketLoggingRequest</a> 。



表 16-2 SetBucketLoggingRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释：</b> 产生日志的桶名称。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值：</b> 无</p>
loggingConfiguration	<a href="#">BucketLoggingConfiguration</a>	必选	<p><b>参数解释：</b> 桶日志配置，详见<a href="#">BucketLoggingConfiguration</a>。</p>

表 16-3 BucketLoggingConfiguration

参数名称	参数类型	是否必选	描述
agency	String	如果是设置桶日志配置则必选	<p><b>参数解释:</b> 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<a href="#">创建 IAM 委托</a>。</p> <p><b>约束限制:</b> 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><b>默认取值:</b> 无</p>
targetBucket Name	String	可选	<p><b>参数解释:</b> 日志存储桶的桶名，该桶用于存放日志文件。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>需要和开启日志功能的桶在同一个 region 中。</li> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63 个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用 IP 地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my.bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-.bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
logfilePrefix	String	可选	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
targetGrantsList	List<GrantAndPermission>	可选	<p><b>参数解释:</b> 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<a href="#">GrantAndPermission</a>。</p>

表 16-4 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	GranteeInterface	必选	<p><b>参数解释:</b> 被授权用户或用户组，详见<a href="#">GranteeInterface</a>。</p>
permission	Permission	必选	<p><b>参数解释:</b> 用户或用户组被授予的权限。</p> <p><b>取值范围:</b> 详见<a href="#">Permission</a>。</p> <p><b>默认取值:</b> 无</p>
delivered	boolean	可选	<p><b>参数解释:</b> 桶的ACL是否向桶内对象传递，作用于桶内所有对象。</p> <p><b>取值范围:</b> true: 是，桶ACL向桶内对象传递。 false: 否，桶ACL不向桶内对象传递，仅作用于桶。</p> <p><b>默认取值:</b> false</p>

表 16-5 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 16-6 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 16-7 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 16-8 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。
PERMISSION_WRITE_ACP	WRITE_ACP	修改ACL配置的权限。 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。
PERMISSION_FULL_CONTROL	FULL_CONTROL	完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。 如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。 如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。

## 返回结果说明

表 16-9 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例：开启桶日志

本示例用于设置examplebucket桶的日志管理配置，委托名称为your agency，生成以targetprefix为前缀的日志文件，并将此日志存放在targetbucketname桶内。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketLoggingConfiguration;
public class SetBucketLogging001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 开启桶日志
            BucketLoggingConfiguration config = new BucketLoggingConfiguration();
```

```
// 设置委托名字, 需要到统一身份认证服务 ( IAM ) 创建委托
config.setAgency("your agency");
config.setTargetBucketName("targetbucketname");
config.setLogFilePrefix("targetprefix");
obsClient.setBucketLogging("examplebucket", config);
System.out.println("setBucketLogging successfully");
} catch (ObsException e) {
    System.out.println("setBucketLogging failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("setBucketLogging failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：为日志对象设置读权限

本示例用于设置examplebucket桶的日志管理配置，委托名称为your agency，生成以prefix为前缀的日志文件，将此日志存放在targetbucketname桶内，并为所有用户设置日志对象的读权限。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketLoggingConfiguration;
import com.obs.services.model.GrantAndPermission;
import com.obs.services.model.GroupGrantee;
import com.obs.services.model.Permission;
public class SetBucketLogging002
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint, 此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 为日志对象设置权限
            String targetBucket = "targetbucketname";
            // 设置桶的日志配置
            BucketLoggingConfiguration config = new BucketLoggingConfiguration();
            // 设置委托名字, 需要到统一身份认证服务 ( IAM ) 创建委托
            config.setAgency("your agency");
```

```
        config.setTargetBucketName(targetBucket);
        config.setLogfilePrefix("prefix");
        // 为所有用户设置对日志对象的读权限
        GrantAndPermission grant1 = new GrantAndPermission(GroupGrantee.ALL_USERS,
Permission.PERMISSION_READ);
        config.setTargetGrants(new GrantAndPermission[]{grant1});
        obsClient.setBucketLogging("examplebucket", config);
        System.out.println("setBucketLogging successfully");
    } catch (ObsException e) {
        System.out.println("setBucketLogging failed");
        // 请求失败,打印http状态码
        System.out.println("HTTP Code:" + e.getResponseCode());
        // 请求失败,打印服务端错误码
        System.out.println("Error Code:" + e.getErrorCode());
        // 请求失败,打印详细错误信息
        System.out.println("Error Message:" + e.getErrorMessage());
        // 请求失败,打印请求id
        System.out.println("Request ID:" + e.getErrorRequestId());
        System.out.println("Host ID:" + e.getErrorHostId());
        e.printStackTrace();
    } catch (Exception e) {
        System.out.println("setBucketLogging failed");
        // 其他异常信息打印
        e.printStackTrace();
    }
}
```

## 代码示例：关闭桶日志功能

关闭桶日志功能实际上就是调用ObsClient.setBucketLogging将examplebucket桶的日志配置清空，以下代码展示了如何关闭桶日志。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketLoggingConfiguration;
public class SetBucketLogging003
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 对桶设置空的日志配置来关闭桶日志
            obsClient.setBucketLogging("examplebucket", new BucketLoggingConfiguration());
            System.out.println("setBucketLogging successfully");
        } catch (ObsException e) {
            System.out.println("setBucketLogging failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
        }
    }
}
```



```
// 请求失败,打印详细错误信息
System.out.println("Error Message:" + e.getErrorMessage());
// 请求失败,打印请求id
System.out.println("Request ID:" + e.getErrorRequestId());
System.out.println("Host ID:" + e.getErrorHostId());
e.printStackTrace();
} catch (Exception e) {
    System.out.println("setBucketLogging failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于开启桶日志的API说明, 请参见[设置桶日志管理配置](#)。
- 更多关于开启桶日志的示例代码, 请参见[Github示例](#)。
- 开启桶日志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](#)。

## 16.3 获取桶日志规则(Java SDK)

### 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。

#### 须知

开发过程中, 您有任何问题可以在github上[提交issue](#), 或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 您必须是桶拥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以及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(final [BaseBucketRequest request](#))

## 请求参数说明

表 16-10 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">BaseBucketRequest</a>	必选	<b>参数解释:</b> 查看桶日志配置请求参数, 详见 <a href="#">BaseBucketRequest</a> 。

表 16-11 BaseBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释:</b> 桶名。 <b>约束限制:</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下:<ul style="list-style-type: none"><li>3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻(如: “my..bucket”)。</li><li>禁止“.”和“-”相邻(如: “my-.bucket”和“my.bucket”)。</li></ul></li><li>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。</li></ul> <b>默认取值:</b> 无

## 返回结果说明

表 16-12 BucketLoggingConfiguration

参数名称	参数类型	是否必选	描述
agency	String	如果是设置桶日志配置则必选	<p><b>参数解释：</b></p> <p>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>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<a href="#">创建IAM委托</a>。</p> <p><b>约束限制：</b></p> <p>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><b>默认取值：</b></p> <p>无</p>

参数名称	参数类型	是否必选	描述
targetBucket Name	String	可选	<p><b>参数解释:</b> 日志存储桶的桶名, 该桶用于存放日志文件。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>● 需要和开启日志功能的桶在同一个 region 中。</li><li>● 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。</li><li>● 桶命名规则如下:<ul style="list-style-type: none"><li>- 3~63个字符, 数字或字母开头, 支持小写字母、数字、“-”、“.”。</li><li>- 禁止使用IP地址。</li><li>- 禁止以“-”或“.”开头及结尾。</li><li>- 禁止两个“.”相邻(如: “my.bucket”)。</li><li>- 禁止“.”和“-”相邻(如: “my-.bucket”和“my.-bucket”)。</li></ul></li><li>● 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。</li></ul> <p><b>默认取值:</b> 无</p>
logfilePrefix	String	可选	<p><b>参数解释:</b> 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。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
targetGrants List	List<GrantAndPermission>	可选	<p><b>参数解释:</b> 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 <a href="#">GrantAndPermission</a>。</p>

表 16-13 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组, 详见 <a href="#">GranteeInterface</a> 。
permission	<a href="#">Permission</a>	必选	<b>参数解释:</b> 用户或用户组被授予的权限。 <b>取值范围:</b> 详见 <a href="#">Permission</a> 。 <b>默认取值:</b> 无
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false

表 16-14 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 16-15 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<p><b>参数解释:</b> 被授权用户的账号ID, 即domain_id。</p> <p><b>取值范围:</b> 如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</p> <p><b>默认取值:</b> 无</p>
displayName	String	可选	<p><b>参数描述:</b> 被授权用户的账号名。</p> <p><b>取值范围:</b> 如何获取账号名请参见<a href="#">如何获取账号名?</a>。</p> <p><b>默认取值:</b> 无</p>

表 16-16 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 16-17 Permission

常量名	原始值	说明
PERMISSION_READ	READ	<p>读权限。</p> <p>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。</p> <p>如果有对象的读权限, 则可以获取该对象内容和元数据。</p>
PERMISSION_WRITE	WRITE	<p>写权限。</p> <p>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。</p> <p>此权限在对象上不适用。</p>

常量名	原始值	说明
PERMISSION_READ_ACP	READ_ACP	<p>读取ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有读对应桶或对象ACP的权限。</p>
PERMISSION_WRITE_ACP	WRITE_ACP	<p>修改ACL配置的权限。</p> <p>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）。</p> <p>桶或对象的所有者永远拥有写对应桶或对象的ACP的权限。</p> <p>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。</p>
PERMISSION_FULL_CONTROL	FULL_CONTROL	<p>完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。</p> <p>如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。</p> <p>如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。</p>

## 代码示例

本示例用于获取examplebucket桶的日志管理配置。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketLoggingConfiguration;
public class GetBucketLogging001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
```

```
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 查看桶日志配置
    BucketLoggingConfiguration config = obsClient.getBucketLogging("examplebucket");
    System.out.println("TargetBucketName:" + config.getTargetBucketName());
    System.out.println("LogfilePrefix:" + config.getLogfilePrefix());
    System.out.println("getBucketLogging successfully");
} catch (ObsException e) {
    System.out.println("getBucketLogging failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getBucketLogging failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于查看桶日志配置的API说明, 请参见[获取桶日志管理配置](#)。
- 更多关于查看桶日志配置的示例代码, 请参见[Github示例](#)。
- 查看桶日志配置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](#)。



# 17 静态网站托管(Java SDK)

## 17.1 静态网站托管简介(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

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](#)。

## 17.2 网站文件托管(Java SDK)

### 功能说明

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。第三方用户在访问网站的时候，实际上是在访问OBS的桶中的对象。

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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](#)。

- 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以及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

1. 上传对象：  
obsClient.putObject([PutObjectRequest request](#))
2. 设置对象访问权限  
obsClient.setObjectAcl([SetObjectAclRequest acl](#))

## 上传对象请求参数说明

表 17-1 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">PutObjectRequest</a>	必选	<b>参数解释：</b> 上传对象请求参数，详见 <a href="#">PutObjectRequest</a> 。

表 17-2 PutObjectRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
metadata	<b>ObjectMetadata</b>	可选	<p><b>参数解释:</b> 对象元数据，详见<b>ObjectMetadata</b>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 创建对象时可指定的预定义访问策略，您可选的访问策略选项参见<a href="#">AccessControlList</a>，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> 无</p>
sseKmsHeader	<a href="#">SseKmsHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseKmsHeader</a>。</p> <p><b>默认取值:</b> 无</p>
sseCHeader	<a href="#">SseCHeader</a>	可选	<p><b>参数解释:</b> 服务端加密头信息。详见<a href="#">SseCHeader</a>。</p> <p><b>默认取值:</b> 无</p>
input	java.io.InputStream	可选	<p><b>参数解释:</b> 待上传对象的数据流。</p> <p><b>默认取值:</b> 无</p>
file	java.io.File	可选	<p><b>参数解释:</b> 待上传对象的文件。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
extensionPermissionMap	Map< <a href="#">ExtensionObjectPermissionEnum</a> , Set<String>>	可选	<p><b>参数解释:</b> 桶ACL的授权Map，您可以为一个或多个账号授予桶权限。Map的<a href="#">ExtensionObjectPermissionEnum</a>用于指定权限，Map的Set&lt;String&gt;用于说明该权限授予的账号ID列表，即domain_id列表。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>授予权限的范围详见<a href="#">ExtensionObjectPermissionEnum</a>。</li> <li>如何获取账号ID请参见<a href="#">如何获取账号ID和用户ID?</a>。</li> </ul> <p><b>默认取值:</b> 无</p>
expires	int	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 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> <p><b>取值范围:</b> 大于0的整型数，单位：天。</p> <p><b>默认取值:</b> 无</p>
progressListener	<a href="#">ProgressListener</a>	可选	<p><b>参数解释:</b> 上传进度，详见<a href="#">ProgressListener</a>。</p>

参数名称	参数类型	是否必选	描述
encodeHeaders	boolean	可选	<p><b>参数解释:</b> 是否开启OBS对请求头域的自动编码。 由于HTTP编码规范限制,无法发送非ASCII码字符,SDK会在发送请求时对您头域中的<b>中文汉字</b>进行url编码,发送编码后数据。如您设置的值content-disposition为“attachment; filename=“中文.txt””,则对象元数据中存储的信息为“attachment; filename=%E4%B8%AD%E6%96%87.txt”。使用浏览器访问时浏览器将会自动解码。</p> <p><b>取值范围:</b> true: 启用SDK编码。 false: 不启用SDK编码。</p> <p><b>默认取值:</b> true</p>

表 17-3 ProgressListener(传输进度接口)成员如下

方法名称	返回值类型	是否必选	描述
<a href="#">progressChanged</a>	void	必选	<p><b>参数解释:</b> 获取上传进度,详见<a href="#">progressChanged</a>。</p> <p><b>默认取值:</b> 无</p>

表 17-4 progressChanged 参数列表

参数名称	参数类型	是否必选	描述
status	<a href="#">ProgressStatus</a>	必选	<p><b>参数解释:</b> 传输进度数据,详见<a href="#">ProgressStatus</a>。</p> <p><b>默认取值:</b> 无</p>

表 17-5 ProgressStatus 方法

方法名称	返回值类型	说明
getAverageSpeed()	double	上传平均速率。
getInstantaneousSpeed()	double	上传瞬时速率。
getTransferPercentage()	int	上传进度百分比。
getNewlyTransferredBytes()	long	新增的字节数。
getTransferredBytes()	long	已传输的字节数。
getTotalBytes()	long	待传输的字节数。

表 17-6 ExtensionObjectPermissionEnum

常量值	说明
GRANT_READ	授权指定租户有读对象和获取对象元数据的权限。
GRANT_READ_ACP	授权指定租户有获取对象ACL的权限。
GRANT_WRITE_ACP	授权指定租户有写对象ACL的权限。
GRANT_FULL_CONTROL	授权指定租户有读对象、获取对象元数据、获取对象ACL、写对象ACL的权限。

表 17-7 SseKmsHeader

参数名称	参数类型	是否必选	描述
encryption	<a href="#">ServerEncryption</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。</p> <p><b>取值范围:</b> 可选值: kms, 即选择SSE-KMS方式加密对象, 详见<a href="#">ServerEncryption</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持kms。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
kmsKeyId	String	可选	<p><b>参数解释:</b> SSE-KMS加密方式下使用的KMS主密钥的ID值。</p> <p><b>取值范围:</b> 有效值支持两种格式:</p> <ol style="list-style-type: none"><li>regionID:domainID(账号ID):key/key_id</li><li>key_id</li></ol> <p>其中:</p> <ul style="list-style-type: none"><li>regionID是使用密钥所属region的ID,可在<a href="#">地区和终端节点</a>页面获取;</li><li>domainID是使用密钥所属账号的账号ID,获取方法参见<a href="#">如何获取账号ID和用户ID?</a>;</li><li>key_id是从数据加密服务创建的密钥ID,获取方法请参见<a href="#">查看密钥</a>。</li></ul> <p><b>默认取值:</b></p> <ul style="list-style-type: none"><li>如果用户没有提供该头域,那么默认的主密钥将会被使用。</li><li>如果默认主密钥不存在,将默认创建并使用。</li></ul>



表 17-8 SseCHeader

参数名称	参数类型	是否必选	描述
algorithm	<a href="#">ServerAlgorithm</a>	必选	<p><b>参数解释:</b> 表示服务端加密是SSE-C方式，对象使用SSE-C加密方式。</p> <p><b>取值范围:</b> 可选值：AES256，即选择SSE-C方式，使用高级加密标准（Advanced Encryption Standard, AES）加密对象。详见<a href="#">ServerAlgorithm</a>。</p> <p><b>默认取值:</b> 无</p>
sseAlgorithm	<a href="#">SSEAlgorithmEnum</a>	可选	<p><b>参数解释:</b> 加密算法。</p> <p><b>约束限制:</b> 只支持AES256。</p> <p><b>取值范围:</b> 详见<a href="#">SSEAlgorithmEnum</a>。</p> <p><b>默认取值:</b> 无</p>
sseCKey	byte[]	必选	<p><b>参数解释:</b> SSE-C方式下加密使用的原始密钥，byte[]形式，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>
sseCKeyBase64	String	可选	<p><b>参数解释:</b> SSE-C方式下的密钥，由原始密钥经过Base64编码后得到，该密钥用于加密对象。</p> <p><b>默认取值:</b> 无</p>

表 17-9 ServerEncryption

常量值	原始值
OBS_KMS	kms

表 17-10 SSEAlgorithmEnum

常量值	原始值
KMS	kms
AES256	AES256

表 17-11 ServerAlgorithm

常量值	原始值
AES256	AES256

表 17-12 AccessControlList

参数名称	参数类型	是否必选	参数类型
owner	<b>Owner</b>	可选	<b>参数解释:</b> 桶所有者的信息, 详见 <b>Owner</b> 。
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false
grants	Set< <b>GrantAndPermission</b> >	可选	<b>参数解释:</b> 被授权用户相关信息, 详见 <b>GrantAndPermission</b> 。

表 17-13 Owner

参数名称	参数类型	是否必选	描述
id	String	必选	<b>参数解释:</b> 桶所有者的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
display Name	String	可选	<b>参数解释:</b> 所有者的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 17-14 GrantAndPermission

参数名称	参数类型	是否必选	描述
grantee	<a href="#">GranteeInterface</a>	必选	<b>参数解释:</b> 被授权用户或用户组, 详见 <a href="#">GranteeInterface</a> 。
permission	<a href="#">Permission</a>	必选	<b>参数解释:</b> 用户或用户组被授予的权限。 <b>取值范围:</b> 详见 <a href="#">Permission</a> 。 <b>默认取值:</b> 无
delivered	boolean	可选	<b>参数解释:</b> 桶的ACL是否向桶内对象传递, 作用于桶内所有对象。 <b>取值范围:</b> true: 是, 桶ACL向桶内对象传递。 false: 否, 桶ACL不向桶内对象传递, 仅作用于桶。 <b>默认取值:</b> false

表 17-15 Permission

常量名	原始值	说明
PERMISSION_READ	READ	读权限。 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。
PERMISSION_WRITE	WRITE	写权限。 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。
PERMISSION_READ_ACP	READ_ACP	读取ACL配置的权限。 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。
PERMISSION_WRITE_ACP	WRITE_ACP	修改ACL配置的权限。 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。
PERMISSION_FULL_CONTROL	FULL_CONTROL	完全控制权限，包括对桶或对象的读写权限，以及对桶或对象ACL配置的读写权限。 如果有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。 如果有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。

表 17-16 GranteeInterface

参数名称	参数类型	是否必选	描述
<a href="#">CanonicalGrantee</a>	<a href="#">CanonicalGrantee</a>	必选	<b>参数解释:</b> 被授权用户的用户信息, 详见 <a href="#">CanonicalGrantee</a> 。
<a href="#">GroupGrantee</a>	<a href="#">GroupGrantee</a>	必选	<b>参数解释:</b> 被授权用户组的用户组信息。 <b>取值范围:</b> 详见 <a href="#">GroupGrantee</a> 。 <b>默认取值:</b> 无

表 17-17 CanonicalGrantee

参数名称	参数类型	是否必选	描述
grantId	String	如果Type为用户类型则必选	<b>参数解释:</b> 被授权用户的账号ID, 即domain_id。 <b>取值范围:</b> 如何获取账号ID请参见 <a href="#">如何获取账号ID和用户ID?</a> 。 <b>默认取值:</b> 无
displayName	String	可选	<b>参数描述:</b> 被授权用户的账号名。 <b>取值范围:</b> 如何获取账号名请参见 <a href="#">如何获取账号名?</a> 。 <b>默认取值:</b> 无

表 17-18 GroupGrantee

常量名	说明
ALL_USERS	所有用户。
AUTHENTICATED_USERS	授权用户, 已废弃。
LOG_DELIVERY	日志投递组, 已废弃。

表 17-19 ObjectMetadata

参数名称	参数类型	是否必选	描述
contentLength	Long	可选	<p><b>参数解释:</b> 对象数据的长度。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 单次上传对象大小范围是[0, 5GB]。</li> <li>• 如果需要上传超过5GB的大文件，需要通过<a href="#">多段操作</a>来分段上传。</li> </ul> <p><b>默认取值:</b> 如果不设置，则SDK会自动计算对象数据的长度。</p>
contentType	String	可选	<p><b>参数解释:</b> 对象的文件类型（MIME类型）。contentType（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><b>取值范围:</b> 常见的contentType（MIME）列表参见<a href="#">如何理解Content-Type（MIME）？(Java SDK)</a>。</p> <p><b>默认取值:</b> 如果未指定Content-Type，SDK会根据指定Object名称的后缀来判定文件类型并自动填充Content-Type（如.xml判断为application/xml文件；.html判断为text/html文件）。</p>
contentEncoding	String	可选	<p><b>参数解释:</b> 响应中的Content-Encoding头。指定对象内容编码格式。</p> <p><b>默认取值:</b> 无</p>
contentDisposition	String	可选	<p><b>参数解释:</b> 为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示，或在弹出文件下载对话框时显示。</p> <p><b>默认取值:</b> 无</p>
cacheControl	String	可选	<p><b>参数解释:</b> 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><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
contentLanguage	String	可选	<b>参数解释:</b> 说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。 <b>默认取值:</b> 无
expires	String	可选	<b>参数解释:</b> 指对象在网页中的缓存过期时间。 <b>约束限制:</b> 日期格式为GMT的格式。 <b>默认取值:</b> 无
contentMd5	String	可选	<b>参数解释:</b> 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。OBS服务端会将该MD5值与对象数据计算出的MD5值进行对比，如果不匹配则上传失败，返回HTTP 400错误。 <b>约束限制:</b> <ul style="list-style-type: none"><li>对象数据的MD5值必须经过Base64编码。</li><li>如果不设置对象的MD5值，OBS服务端会忽略对对象数据的MD5值校验。</li></ul> <b>取值范围:</b> 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：n58lG6hfM7vql4K0vnWpog== <b>默认取值:</b> 无
storageClasses	<a href="#">StorageClassEnum</a>	可选	<b>参数解释:</b> 对象的存储类别。创建对象时，可以加上此头域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。 <b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
webSiteRedirectLocation	String	可选	<p><b>参数解释:</b> 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，该参数指明对象的重定向地址。</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/</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 必须以“/”、“http://”或“https://”开头，长度不超过2KB。</li> <li>• 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。</li> </ul> <p><b>默认取值:</b> 无</p>
nextPosition	long	可选	<p><b>参数解释:</b> 下次追加上传的位置。</p> <p><b>取值范围:</b> 0~对象长度，单位：字节。</p> <p><b>默认取值:</b> 无</p>
appendable	boolean	可选	<p><b>参数解释:</b> 对象是否为可追加上传的对象。</p> <p><b>取值范围:</b> true: 是，可追加上传。 false: 否，不可追加上传。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	是否必选	描述
userMetadata	Map<String, Object>	可选	<p><b>参数解释:</b> 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。Map中的String代表以“x-obs-meta-”开头的自定义元数据名称，Object代表自定义元数据的值。对象的自定义元数据可以通过ObsClient.getObjectMetadata获取，参见<a href="#">获取对象元数据</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 一个对象可以有多个元数据，总大小不能超过8KB。</li> <li>• 使用ObsClient.getObject下载对象时，对象的自定义元数据也会同时下载。</li> </ul> <p><b>默认取值:</b> 无</p>

表 17-20 StorageClassEnum

常量名	原始值	说明
STANDARD	STANDARD	标准存储。
WARM	WARM	低频访问存储。
COLD	COLD	归档存储。

表 17-21 ACL 预定义访问策略

常量名	说明
AccessControlList.REST_CANNED_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。
AccessControlList.REST_CANNED_PUBLIC_READ	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。 设在对象上，所有人可以获取该对象内容和元数据。

常量名	说明
AccessControllist.REST_CAN NED_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、上传对象删除对象、初始化段任务、上传段、合并段、拷贝段、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。
AccessControllist.REST_CAN NED_PUBLIC_READ_DELIVER ED	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，可以获取该桶内对象的内容和元数据。 不能应用于对象。
AccessControllist.REST_CAN NED_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、上传对象、删除对象、初始化段任务、上传段、合并段、拷贝段、取消多段上传任务，可以获取该桶内对象的内容和元数据。 不能应用于对象。

## 设置对象访问权限请求参数说明

表 17-22 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">SetObjectAclRequest</a>	必选	<b>参数解释：</b> 设置对象ACL请求参数，详见 <a href="#">SetObjectAclRequest</a> 。

表 17-23 SetObjectAclRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my.bucket”）。</li><li>禁止“.”和“-”相邻（如：“my.bucket”和“my.bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>
objectKey	String	必选	<p><b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为 folder/test.txt。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
versionId	String	可选	<p><b>参数解释:</b> 对象的版本号。</p> <p><b>取值范围:</b> 长度为32的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
acl	<a href="#">AccessControlList</a>	可选	<p><b>参数解释:</b> 指定对象的ACL访问策略，您可以使用预定义的ACL策略，也可以自定义ACL策略，有关访问控制列表（Access Control List, ACL）功能的详细信息可参见<a href="#">ACL功能介绍</a>。</p> <p><b>取值范围:</b></p> <ul style="list-style-type: none"> <li>• 如果使用预定义ACL策略，则可选策略参见<a href="#">ACL预定义访问策略</a>。</li> <li>• 如果选择自定义ACL策略，则可以根据<a href="#">AccessControlList</a>参数描述，自行给参数赋值。</li> </ul> <p><b>默认取值:</b> AccessControlList.REST_CANNED_PRIVATE</p>

## 上传对象返回结果说明

表 17-24 PutObjectResult

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> 响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
storageClass	<a href="#">StorageClassEnum</a>	<b>参数解释:</b> 对象的存储类别。当对象存储类别是标准存储时，该值为空。 <b>取值范围:</b> 可选择的存储类别参见 <a href="#">StorageClassEnum</a> 。 <b>默认取值:</b> 无
versionId	String	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
etag	String	<b>参数解释:</b> 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。上传的对象或拷贝操作创建的对象，都有唯一的etag。 <b>约束限制:</b> 当对象是服务端加密的对象时，etag值不是对象的MD5值。 <b>取值范围:</b> 长度为32的字符串。 <b>默认取值:</b> 无
objectKey	String	<b>参数解释:</b> 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.cn-north-4.myhuaweicloud.com/folder/test.txt 中，对象名为folder/test.txt。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无

参数名称	参数类型	描述
bucketName	String	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>

## 设置对象访问权限返回结果说明

表 17-25 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

以下代码展示了examplebucket桶如何实现网站文件托管：（1）上传对象时，设置对象的MIME类型；（2）设置对象访问权限为公共读。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.AccessControlList;
import com.obs.services.model.ObjectMetadata;
import com.obs.services.model.PutObjectRequest;
import java.io.File;
public class PutObject001
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 以下代码展示了如何实现网站文件托管
            // 上传对象，设置对象MIME类型
            PutObjectRequest request = new PutObjectRequest();
            request.setBucketName("examplebucket");
            request.setObjectKey("test.html");
            request.setFile(new File("localfile.html"));
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.setContentType("text/html");
            request.setMetadata(metadata);
            obsClient.putObject(request);
            // 设置对象访问权限为公共读
            obsClient.setObjectAcl("examplebucket", "test.html",
            AccessControlList.REST_CANNED_PUBLIC_READ);
            System.out.println("putObject successfully");
        } catch (ObsException e) {
            System.out.println("putObject failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("putObject failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

```
}  
}
```

#### 📖 说明

上例中您可以使用<https://bucketname.your-endpoint/test.html>在浏览器直接访问托管的文件。

## 相关链接

- 关于上传对象的API说明，请参见[POST上传](#)。
- 关于设置对象ACL的API说明，请参见[设置对象ACL](#)。
- 网站文件托管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](#)。

## 17.3 设置托管配置(Java SDK)

### 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。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及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(final SetBucketWebsiteRequest request)
```



## 请求参数说明

表 17-26 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">SetBucketWebsiteRequest</a>	必选	<b>参数解释：</b> 设置托管配置请求参数，详见 <a href="#">SetBucketWebsiteRequest</a> 。

表 17-27 SetBucketWebsiteRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释：</b> 桶名。 <b>约束限制：</b> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <b>取值范围：</b> 长度为3~63个字符。
websiteConfig	<a href="#">WebsiteConfiguration</a>	必选	<b>参数解释：</b> 静态网站设置托管配置信息，详见 <a href="#">WebsiteConfiguration</a> 。

表 17-28 WebsiteConfiguration

参数名称	参数类型	是否必选	描述
suffix	String	可选	<p><b>参数解释:</b> 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><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>key、suffix和routeRules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key、suffix和routeRules。</li> <li>当key、suffix和routeRules三个字段一起使用时，routeRules可为空。</li> <li>设置key、suffix和routeRules时不能全为空。</li> </ul> <p><b>取值范围:</b> 该字段不能为空或者包含“/”字符。</p> <p><b>默认取值:</b> 无</p>
key	String	可选	<p><b>参数解释:</b> 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><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>key、suffix和routeRules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key、suffix和routeRules。</li> <li>当key、suffix和routeRules三个字段一起使用时，routeRules可为空。</li> <li>设置key、suffix和routeRules时不能全为空。</li> </ul> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
routeRules	List< <a href="#">Route Rule</a> >	可选	<p><b>参数解释:</b> 请求重定向规则列表, 详见<a href="#">RouteRule</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>key、suffix和routeRules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key、suffix和routeRules。</li><li>当key、suffix和routeRules三个字段一起使用时, routeRules可为空。</li><li>设置key、suffix和routeRules时不能全为空。</li></ul> <p><b>默认取值:</b> 无</p>
redirectAllRequestsTo	<a href="#">RedirectAllRequest</a>	可选	<p><b>参数解释:</b> 所有请求重定向规则, 详见<a href="#">RedirectAllRequest</a>。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>key、suffix和routeRules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key、suffix和routeRules。</li><li>设置redirectAllRequestsTo时不能为空。</li></ul> <p><b>默认取值:</b> 无</p>

表 17-29 RedirectAllRequest

参数名称	参数类型	是否必选	描述
hostName	String	必选	<p><b>参数解释:</b> 重定向时使用的域名。如 www.example.com。</p> <p><b>约束限制:</b> 域名需符合域名规范。</p> <p><b>默认取值:</b> 无</p>
protocol	<a href="#">ProtocolEnum</a>	可选	<p><b>参数解释:</b> 重定向时使用的协议。</p> <p><b>取值范围:</b> 详见<a href="#">ProtocolEnum</a>。</p> <p><b>默认取值:</b> 无</p>

表 17-30 ProtocolEnum

常量名	原始值	说明
HTTP	http	重定向请求时使用HTTP协议。
HTTPS	https	重定向请求时使用HTTPS协议。

表 17-31 RouteRule

参数名称	参数类型	是否必选	描述
condition	<a href="#">RouteRuleCondition</a>	可选	<p><b>参数解释:</b> 重定向规则的匹配条件。详见<a href="#">RouteRuleCondition</a>。</p>
redirect	<a href="#">Redirect</a>	必选	<p><b>参数解释:</b> 重定向请求时的具体信息。详见<a href="#">Redirect</a>。</p>

表 17-32 RouteRuleCondition

参数名称	参数类型	是否必选	描述
keyPrefixEquals	String	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 与httpErrorCodeReturnedEquals参数不可同时使用, 两者二选一。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
httpErrorCodeReturnedEquals	String	可选	<p><b>参数解释:</b> 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RouteRuleCondition中的httpErrorCodeReturnedEquals设置为404, Redirect中的replaceKeyWith设置为NotFound.html。</p> <p><b>约束限制:</b> 与keyPrefixEquals参数不可同时使用, 两者二选一。</p> <p><b>取值范围:</b> 取值范围可参见<a href="#">错误码</a>。</p> <p><b>默认取值:</b> 无</p>

表 17-33 Redirect

参数名称	参数类型	是否必选	描述
Protocol	Protocol Enum	可选	<b>参数解释:</b> 重定向时使用的协议。 <b>取值范围:</b> 详见ProtocolEnum。 <b>默认取值:</b> 无
hostName	String	可选	<b>参数解释:</b> 重定向请求时使用的域名。 <b>默认取值:</b> 无
replaceKeyPrefix With	String	可选	<b>参数解释:</b> 重定向请求时使用的对象名前缀。 <b>约束限制:</b> 不可与replaceKeyWith同时使用。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无
replaceKeyWith	String	可选	<b>参数解释:</b> 重定向请求时使用的对象名。 <b>约束限制:</b> 不可与replaceKeyPrefixWith同时使用。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无
httpRedirectCode	String	可选	<b>参数解释:</b> 重定向请求时响应中的HTTP状态码，详见状态码。 <b>默认取值:</b> 无

## 返回结果说明

表 17-34 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例：配置默认主页和错误页面

以下代码展示了如何为examplebucket桶配置默认主页和错误页面。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.WebsiteConfiguration;
public class SetBucketWebsite001
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 以下代码展示了如何配置默认主页和错误页面
            WebsiteConfiguration config = new WebsiteConfiguration();
```

```
// 配置默认主页
config.setSuffix("index.html");
// 配置错误页面
config.setKey("error.html");
obsClient.setBucketWebsite("examplebucket", config);
System.out.println("setBucketWebsite successfully");
} catch (ObsException e) {
    System.out.println("setBucketWebsite failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("setBucketWebsite failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：配置重定向规则

以下代码展示了如何为examplebucket桶配置重定向规则。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ProtocolEnum;
import com.obs.services.model.Redirect;
import com.obs.services.model.RouteRule;
import com.obs.services.model.RouteRuleCondition;
import com.obs.services.model.WebsiteConfiguration;
public class SetBucketWebsite002
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 以下代码展示了如何配置重定向规则
            WebsiteConfiguration config = new WebsiteConfiguration();
            // 配置默认主页
            config.setSuffix("index.html");
            // 配置错误页面
            config.setKey("error.html");
            RouteRule rule = new RouteRule();
            Redirect r = new Redirect();
```



```
r.setHostName("www.example.com");
r.setHttpRedirectCode("305");
r.setRedirectProtocol(ProtocolEnum.HTTP);
r.setReplaceKeyPrefixWith("replacekeyprefix");
rule.setRedirect(r);
RouteRuleCondition condition = new RouteRuleCondition();
condition.setHttpErrorCodeReturnedEquals("404");
condition.setKeyPrefixEquals("keyprefix");
rule.setCondition(condition);
config.getRouteRules().add(rule);
obsClient.setBucketWebsite("examplebucket", config);
System.out.println("setBucketWebsite successfully");
} catch (ObsException e) {
    System.out.println("setBucketWebsite failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("setBucketWebsite failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 代码示例：配置所有请求重定向

以下代码展示了如何为examplebucket桶配置所有请求重定向。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.ProtocolEnum;
import com.obs.services.model.RedirectAllRequest;
import com.obs.services.model.WebsiteConfiguration;
public class SetBucketWebsite003
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk, endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 以下代码展示了如何配置所有请求重定向
            WebsiteConfiguration config = new WebsiteConfiguration();
            RedirectAllRequest redirectAll = new RedirectAllRequest();
            redirectAll.setHostName("www.example.com");
```

```
redirectAll.setRedirectProtocol(ProtocolEnum.HTTP);
config.setRedirectAllRequestsTo(redirectAll);
obsClient.setBucketWebsite("examplebucket", config);
System.out.println("setBucketWebsite successfully");
} catch (ObsException e) {
    System.out.println("setBucketWebsite failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("setBucketWebsite failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置桶的托管配置的API说明, 请参见[设置桶的网站配置](#)。
- 更多关于设置桶的托管配置的示例代码, 请参见[Github示例](#)。
- 设置桶的托管配置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](#)。
- 静态网站托管相关常见问题请参见[静态网站托管相关常见问题](#)。

## 17.4 获取托管配置(Java SDK)

### 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。

#### 须知

开发过程中, 您有任何问题可以在github上[提交issue](#), 或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及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(final [BaseBucketRequest request](#))

## 请求参数说明

表 17-35 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">BaseBucketRequest</a>	必选	<b>参数解释：</b> 桶基本信息参数列表，详见 <a href="#">BaseBucketRequest</a> 。

表 17-36 BaseBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释：</b> 桶名。</p> <p><b>约束限制：</b></p> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 桶命名规则如下： <ul style="list-style-type: none"> <li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>- 禁止使用IP地址。</li> <li>- 禁止以“-”或“.”开头及结尾。</li> <li>- 禁止两个“.”相邻（如：“my..bucket”）。</li> <li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>• 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。</li> </ul> <p><b>默认取值：</b> 无</p>

## 返回结果说明

表 17-37 WebsiteConfiguration

参数名称	参数类型	是否必选	描述
suffix	String	可选	<b>参数解释:</b> 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）。 <b>取值范围:</b> 该字段不能为空或者包含“/”字符。 <b>默认取值:</b> 无
key	String	可选	<b>参数解释:</b> 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。 <b>取值范围:</b> 长度大于0且不超过1024的字符串。 <b>默认取值:</b> 无
redirectAllRequestsTo	<a href="#">RedirectAllRequest</a>	可选	<b>参数解释:</b> 所有请求重定向规则，详见 <a href="#">RedirectAllRequest</a> 。
routeRules	List< <a href="#">RouteRule</a> >	可选	<b>参数解释:</b> 请求重定向规则列表，详见 <a href="#">RouteRule</a> 。

表 17-38 RedirectAllRequest

参数名称	参数类型	是否必选	描述
hostName	String	必选	<b>参数解释:</b> 重定向时使用的域名。如www.example.com。 <b>约束限制:</b> 域名需符合域名规范。 <b>默认取值:</b> 无

参数名称	参数类型	是否必选	描述
protocol	<a href="#">ProtocolEnum</a>	可选	<b>参数解释:</b> 重定向时使用的协议。 <b>取值范围:</b> 详见 <a href="#">ProtocolEnum</a> 。 <b>默认取值:</b> 无

表 17-39 ProtocolEnum

常量名	原始值	说明
HTTP	http	重定向请求时使用HTTP协议。
HTTPS	https	重定向请求时使用HTTPS协议。

表 17-40 RouteRule

参数名称	参数类型	是否必选	描述
condition	<a href="#">RouteRuleCondition</a>	可选	<b>参数解释:</b> 重定向规则的匹配条件。详见 <a href="#">RouteRuleCondition</a> 。
redirect	<a href="#">Redirect</a>	必选	<b>参数解释:</b> 重定向请求时的具体信息。详见 <a href="#">Redirect</a> 。

表 17-41 RouteRuleCondition

参数名称	参数类型	是否必选	描述
keyPrefixEquals	String	可选	<p><b>参数解释:</b> 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><b>约束限制:</b> 与httpErrorCodeReturnedEquals参数不可同时使用, 两者二选一。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
httpErrorCodeReturnedEquals	String	可选	<p><b>参数解释:</b> 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RouteRuleCondition中的httpErrorCodeReturnedEquals设置为404, Redirect中的replaceKeyWith设置为NotFound.html。</p> <p><b>约束限制:</b> 与keyPrefixEquals参数不可同时使用, 两者二选一。</p> <p><b>取值范围:</b> 取值范围可参见<a href="#">错误码</a>。</p> <p><b>默认取值:</b> 无</p>

表 17-42 Redirect

参数名称	参数类型	是否必选	描述
Protocol	ProtocolEnum	可选	<p><b>参数解释:</b> 重定向时使用的协议。</p> <p><b>取值范围:</b> 详见ProtocolEnum。</p> <p><b>默认取值:</b> 无</p>
hostName	String	可选	<p><b>参数解释:</b> 重定向请求时使用的域名。</p> <p><b>默认取值:</b> 无</p>
replaceKeyPrefixWith	String	可选	<p><b>参数解释:</b> 重定向请求时使用的对象名前缀。</p> <p><b>约束限制:</b> 不可与replaceKeyWith同时使用。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
replaceKeyWith	String	可选	<p><b>参数解释:</b> 重定向请求时使用的对象名。</p> <p><b>约束限制:</b> 不可与replaceKeyPrefixWith同时使用。</p> <p><b>取值范围:</b> 长度大于0且不超过1024的字符串。</p> <p><b>默认取值:</b> 无</p>
httpRedirectCode	String	可选	<p><b>参数解释:</b> 重定向请求时响应中的HTTP状态码，详见状态码。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

您可以通过obsClient.getBucketWebsite查看examplebucket桶的托管配置。以下代码展示了如何查看托管配置。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.RouteRule;
import com.obs.services.model.WebsiteConfiguration;
public class GetBucketWebsite001
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 查看桶的托管配置
            WebsiteConfiguration config = obsClient.getBucketWebsite("examplebucket");
            System.out.println("Key:" + config.getKey());
            System.out.println("Suffix:" + config.getSuffix());
            for(RouteRule rule : config.getRouteRules()){
                System.out.println("rule:" +rule);
            }
            System.out.println("getBucketWebsite successfully");
        } catch (ObsException e) {
            System.out.println("getBucketWebsite failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("getBucketWebsite failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于查看桶的托管配置的API说明，请参见[获取桶的网站配置](#)。
- 更多关于查看桶的托管配置的示例代码，请参见[Github示例](#)。
- 查看桶的托管配置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](#)。
- 静态网站托管相关常见问题请参见[静态网站托管相关常见问题](#)。



## 17.5 删除托管配置(Java SDK)

### 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。

#### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 您必须是桶拥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以及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(final [BaseBucketRequest request](#))

### 请求参数说明

表 17-43 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">BaseBucketRequest</a>	必选	<b>参数解释：</b> 桶基本信息参数列表，详见 <a href="#">BaseBucketRequest</a> 。

表 17-44 BaseBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“_”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 17-45 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

您可以通过obsClient.deleteBucketWebsite删除examplebucket桶的托管配置。以下代码展示了如何删除托管配置。

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class DeleteBucketWebsite001
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 清除桶的托管配置
            obsClient.deleteBucketWebsite("examplebucket");
            System.out.println("deleteBucketWebsite successfully");
        } catch (ObsException e) {
            System.out.println("deleteBucketWebsite failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("deleteBucketWebsite failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于删除桶的托管配置的API说明，请参见[删除桶的网站配置](#)。
- 更多关于删除桶的托管配置的示例代码，请参见[Github示例](#)。
- 删除桶的托管配置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](#)。
- 静态网站托管相关常见问题请参见[静态网站托管相关常见问题](#)。

# 18 桶标签管理(Java SDK)

## 18.1 桶标签简介(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

标签用于标识OBS中的桶，以此来达到对OBS中的桶进行分类的目的。

更多关于桶标签的内容请参考[标签](#)。

## 18.2 设置桶标签(Java SDK)

### 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。

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 接口约束

- 每个桶最多能设置10个标签。

- 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以及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(final [SetBucketTaggingRequest request](#))

## 请求参数说明

表 18-1 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">SetBucketTaggingRequest</a>	必选	<b>参数解释：</b> 设置桶标签请求参数，详见 <a href="#">SetBucketTaggingRequest</a> 。

表 18-2 SetBucketTaggingRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>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。</li> <li>桶命名规则如下： <ul style="list-style-type: none"> <li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>禁止使用IP地址。</li> <li>禁止以“-”或“.”开头及结尾。</li> <li>禁止两个“.”相邻（如：“my..bucket”）。</li> <li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>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。</li> </ul> <p><b>默认取值:</b> 无</p>
bucketTagInfo	<a href="#">BucketTagInfo</a>	必选	<p><b>参数解释:</b> 桶标签信息，详见<a href="#">BucketTagInfo</a>。</p>

表 18-3 BucketTagInfo

参数名称	参数类型	是否必选	描述
tagSet	<a href="#">TagSet</a>	必选	<p><b>参数解释:</b> 标签集，详见<a href="#">TagSet</a>。</p>

表 18-4 TagSet

参数名称	参数类型	是否必选	描述
tags	List< <a href="#">Tag</a> >	必选	<p><b>参数解释:</b> 标签列表，标签信息详见<a href="#">Tag</a>。</p>

表 18-5 Tag

参数名称	参数类型	是否必选	描述
key	String	必选	<p><b>参数解释:</b> 标签的键。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 同一个桶标签的键不能重复。</li><li>• 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。</li><li>• 长度范围为1-36个字符。</li><li>• 支持Unicode。</li><li>• 不能以空格开头或结尾，不能包含字符“ ”、“*”、“ ”、“/”、“&lt;”、“&gt;”、“=”、“\”以及ASCII码0x00--0x1F这些控制字符。</li><li>• 区分大小写。</li></ul> <p><b>默认取值:</b> 无</p>
value	String	必选	<p><b>参数解释:</b> 标签的值。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>• 标签的值可以重复，并且可以为空。</li><li>• 长度范围为0-43个字符。</li><li>• 支持Unicode。</li><li>• 不能包含字符“,”、“*”、“ ”、“/”、“&lt;”、“&gt;”、“=”、“\”以及ASCII码0x00--0x1F这些控制字符。</li><li>• 区分大小写。</li></ul> <p><b>默认取值:</b> 无</p>



## 返回结果说明

表 18-6 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于通过ObsClient.setBucketTagging设置examplebucket桶的桶标签。代码展示如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketTagInfo;
public class SetBucketTagging001
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 设置桶标签

```

```
BucketTagInfo bucketTagInfo = new BucketTagInfo();
BucketTagInfo.TagSet tagSet = new BucketTagInfo.TagSet();
tagSet.addTag("tag1", "value1");
tagSet.addTag("tag2", "value2");
bucketTagInfo.setTagSet(tagSet);
obsClient.setBucketTagging("examplebucket", bucketTagInfo);
System.out.println("setBucketTagging successfully");
} catch (ObsException e) {
    System.out.println("setBucketTagging failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("setBucketTagging failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于设置桶标签的API说明, 请参见[设置桶标签](#)。
- 更多关于设置桶标签的示例代码, 请参见[Github示例](#)。
- 设置桶标签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](#)。
- 标签相关常见问题请参见[标签相关常见问题](#)。

## 18.3 获取桶标签(Java SDK)

### 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。

#### 须知

开发过程中, 您有任何问题可以在github上[提交issue](#), 或者在[华为云对象存储服务论坛](#)中发帖求助。

### 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以及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(final [BaseBucketRequest request](#))

## 请求参数说明

表 18-7 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">BaseBucketRequest</a>	必选	<b>参数解释:</b> 桶基本信息参数列表，详见 <a href="#">BaseBucketRequest</a> 。

表 18-8 BaseBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<b>参数解释:</b> 桶名。 <b>约束限制:</b> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 桶命名规则如下： <ul style="list-style-type: none"> <li>- 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li> <li>- 禁止使用IP地址。</li> <li>- 禁止以“-”或“.”开头及结尾。</li> <li>- 禁止两个“.”相邻（如：“my..bucket”）。</li> <li>- 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li> </ul> </li> <li>• 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。</li> </ul> <b>默认取值:</b> 无

## 返回结果说明

表 18-9 BucketTagInfo

参数名称	参数类型	描述
tagSet	TagSet	<b>参数解释:</b> 标签集, 详见TagSet。
statusCode	int	<b>参数解释:</b> HTTP状态码。 <b>取值范围:</b> 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。 <b>默认取值:</b> 无
responseHeaders	Map<String, Object>	<b>参数解释:</b> HTTP响应消息头列表, 由多个元组构成。元组中String代表响应消息头的名称, Object代表响应消息头的值。 <b>默认取值:</b> 无

表 18-10 TagSet

参数名称	参数类型	描述
tags	List<Tag>	<b>参数解释:</b> 标签列表, 标签信息详见Tag。

表 18-11 Tag

参数名称	参数类型	是否必选	描述
key	String	必选	<p><b>参数解释:</b> 标签的键。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 同一个桶标签的键不能重复。</li> <li>• 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。</li> <li>• 长度范围为1-36个字符。</li> <li>• 支持Unicode。</li> <li>• 不能以空格开头或结尾，不能包含字符“ ”、“*”、“ ”、“/”、“&lt;”、“&gt;”、“=”、“\”以及ASCII码0x00--0x1F这些控制字符。</li> <li>• 区分大小写。</li> </ul> <p><b>默认取值:</b> 无</p>
value	String	必选	<p><b>参数解释:</b> 标签的值。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>• 标签的值可以重复，并且可以为空。</li> <li>• 长度范围为0-43个字符。</li> <li>• 支持Unicode。</li> <li>• 不能包含字符“ ”、“*”、“ ”、“/”、“&lt;”、“&gt;”、“=”、“\”以及ASCII码0x00--0x1F这些控制字符。</li> <li>• 区分大小写。</li> </ul> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于通过obsClient.getBucketTagging获取examplebucket桶的桶标签。代码展示如下：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.BucketTagInfo;
public class GetBucketTagging001
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
```

```
// 【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/SecurityToken，也可以使用其他外部引入方式传入。
// String securityToken = System.getenv("SECURITY_TOKEN");
// endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");

// 创建ObsClient实例
// 使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk,endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 获取桶标签
    BucketTagInfo bucketTagInfo = obsClient.getBucketTagging("examplebucket");
    for(BucketTagInfo.TagSet.Tag tag : bucketTagInfo.getTagSet().getTags()){
        System.out.println("\t" + tag.getKey() + ":" + tag.getValue());
    }
    System.out.println("getBucketTagging successfully");
} catch (ObsException e) {
    System.out.println("getBucketTagging failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("getBucketTagging failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 相关链接

- 关于获取桶标签的API说明，请参见[获取桶标签](#)。
- 更多关于获取桶标签的示例代码，请参见[Github示例](#)。
- 获取桶标签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](#)。
- 标签相关常见问题请参见[标签相关常见问题](#)。

## 18.4 删除桶标签(Java SDK)

### 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。

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

## 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以及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(final [BaseBucketRequest request](#))

## 请求参数说明

表 18-12 请求参数列表

参数名称	参数类型	是否必选	描述
request	<a href="#">BaseBucketRequest</a>	必选	<b>参数解释：</b> 桶基本信息参数列表，详见 <a href="#">BaseBucketRequest</a> 。

表 18-13 BaseBucketRequest

参数名称	参数类型	是否必选	描述
bucketName	String	必选	<p><b>参数解释:</b> 桶名。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"><li>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。</li><li>桶命名规则如下：<ul style="list-style-type: none"><li>3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>禁止使用IP地址。</li><li>禁止以“-”或“.”开头及结尾。</li><li>禁止两个“.”相邻（如：“my..bucket”）。</li><li>禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul></li><li>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。</li></ul> <p><b>默认取值:</b> 无</p>

## 返回结果说明

表 18-14 返回结果 HeaderResponse(SDK 公共响应结果)各项属性说明

参数名称	参数类型	描述
statusCode	int	<p><b>参数解释:</b> HTTP状态码。</p> <p><b>取值范围:</b> 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<a href="#">状态码</a>。</p> <p><b>默认取值:</b> 无</p>



参数名称	参数类型	描述
responseHeaders	Map<String, Object>	<p><b>参数解释:</b> HTTP响应消息头列表，由多个元组构成。元组中String代表响应消息头的名称，Object代表响应消息头的值。</p> <p><b>默认取值:</b> 无</p>

## 代码示例

本示例用于通过ObsClient.deleteBucketTagging删除examplebucket桶的桶标签。以下代码展示了如何删除桶标签：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
public class DeleteBucketTagging001
{
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");

        // 创建ObsClient实例
        // 使用永久AK/SK初始化客户端
        ObsClient obsClient = new ObsClient(ak, sk,endPoint);
        // 使用临时AK/SK和SecurityToken初始化客户端
        // ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

        try {
            // 删除桶标签
            obsClient.deleteBucketTagging("examplebucket");
            System.out.println("deleteBucketTagging successfully");
        } catch (ObsException e) {
            System.out.println("deleteBucketTagging failed");
            // 请求失败,打印http状态码
            System.out.println("HTTP Code:" + e.getResponseCode());
            // 请求失败,打印服务端错误码
            System.out.println("Error Code:" + e.getErrorCode());
            // 请求失败,打印详细错误信息
            System.out.println("Error Message:" + e.getErrorMessage());
            // 请求失败,打印请求id
            System.out.println("Request ID:" + e.getErrorRequestId());
            System.out.println("Host ID:" + e.getErrorHostId());
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("deleteBucketTagging failed");
            // 其他异常信息打印
            e.printStackTrace();
        }
    }
}
```

## 相关链接

- 关于删除桶标签的API说明，请参见[删除桶标签](#)。
- 更多关于删除桶标签的示例代码，请参见[Github示例](#)。
- 删除桶标签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](#)。
- 标签相关常见问题请参见[标签相关常见问题](#)。

# 19 服务端加密(Java SDK)

## 19.1 服务端加密简介(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

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](#)。

## 19.2 加密说明(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

OBS Java SDK支持服务端加密的接口见下表：

OBS Java SDK接口方法	说明	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 SSE-OBS

OBS Java SDK接口方法	说明	支持加密类型
ObsClient.getObject	具有KMS Administrator权限的用户可直接下载KMS加密对象，后端解密后返回。(SSE-KMS) 下载对象时设置解密算法、密钥，用于解密对象。(SSE-C)	SSE-KMS SSE-C SSE-OBS
ObsClient.copyObject	1. 复制对象时设置源对象的解密算法、密钥，用于解密源对象。 2. 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 SSE-OBS
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	1. 复制段时设置源对象的解密算法、密钥，用于解密源对象。 2. 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

## 19.3 加密示例(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### 代码示例：上传对象加密（SSE-C 方式）

以下代码展示了在上传对象时使用SSE-C方式进行服务端加密。

设置SSE-C方式下使用的密钥，请参考[如何生成SSE-C方式的加密密钥](#)。

更多关于服务端加密的内容请参考[服务端加密SSE-C方式](#)。

```
// Endpoint以北京四为例，其他地区请按实际情况填写。  
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";  
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca_01_0003.html  
String ak = System.getenv("ACCESS_KEY_ID");
```

```
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
// 创建ObsClient实例
ObsClient obsClient = new ObsClient(ak, sk, endPoint);

PutObjectRequest request = new PutObjectRequest();
request.setBucketName("bucketname");
request.setObjectKey("objectname");
request.setFile(new File("localfile"));

HashMap<String, String> userHeaders = new HashMap<>();
userHeaders.put("x-obs-server-side-encryption-customer-algorithm", "AES256");
//SSE-C方式下使用该头域, 该头域表示加密对象使用的密钥, 头域值是256位密钥的base64编码。
userHeaders.put("x-obs-server-side-encryption-customer-key", "your-encryption-customer-key");
userHeaders.put("x-obs-server-side-encryption-customer-key-MD5",
    ServiceUtils.toBase64(ServiceUtils.computeMD5Hash(ServiceUtils.fromBase64("your-
encryption-customer-key"))));
request.setUserHeaders(userHeaders);
HeaderResponse response = obsClient.putObject(request);
System.out.println("response:" + response.getRequestId());
```

## 代码示例：上传对象加密（SSE-OBS 方式）

以下代码展示了在上传对象时使用SSE-OBS方式进行服务端加密,通过自定义头域添加头域x-obs-server-side-encryption:AES256实现。

关于服务端加密的用户指南：[服务端加密](#)。

```
// Endpoint以北京四为例, 其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险, 建议在配置文件或者环境变量中密文存
// 放, 使用时解密, 确保安全; 本示例以ak和sk保存在环境变量中为例, 运行本示例前请先在本地环境中设置环境
// 变量ACCESS_KEY_ID和SECRET_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/
// usermanual-ca/ca_01_0003.html
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
// 创建ObsClient实例
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
//设置桶名、对象名、本地文件路径
PutObjectRequest request = new PutObjectRequest();
request.setBucketName("BucketName");
request.setObjectKey("ObjectKey");
request.setFile(new File("LocalFilePath"));
//设置自定义头域, 设置使用sse-obs加密
HashMap<String, String> userHeaders = new HashMap<>();
userHeaders.put("x-obs-server-side-encryption", "AES256");
request.setUserHeaders(userHeaders);
HeaderResponse response = obsClient.putObject(request);
```

## 代码示例：下载对象解密

以下代码展示了在下载对象时使用SSE-C方式进行服务端解密：

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.internal.utils.ServiceUtils;
import com.obs.services.model.GetObjectRequest;
import com.obs.services.model.ObsObject;
import java.util.HashMap;
import java.util.Map;

public class SseCGetObject{
    public static void main(String[] args) {
        // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险, 建议在配置文件或者环境变量中密
        // 文存放, 使用时解密, 确保安全; 本示例以ak和sk保存在环境变量中为例,
        // 运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/
        // usermanual-ca/ca_01_0003.html
```

```
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
// 【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/SecurityToken，也可以使用其他外部引入方式传入。
// String securityToken = System.getenv("SECURITY_TOKEN");
// Endpoint以北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 创建ObsClient实例,使用永久AK/SK初始化客户端
ObsClient obsClient = new ObsClient(ak, sk,endPoint);
// 使用临时AK/SK和SecurityToken初始化客户端
// ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint);

try {
    // 调用接口进行操作，例如下载加密对象
    GetObjectRequest request=new GetObjectRequest("bucketname","objectname");
    // 设置SSE-C算法解密对象
    HashMap<String, String> userHeaders=new HashMap<>();
    userHeaders.put("x-obs-server-side-encryption-customer-algorithm","AES256");
    //SSE-C方式下使用该头域，该头域表示加密对象使用的密钥，头域值是256位密钥的base64编码。
    userHeaders.put("x-obs-server-side-encryption-customer-key","your-encryption-customer-key");
    userHeaders.put("x-obs-server-side-encryption-customer-key-MD5",
        ServiceUtils.toBase64(ServiceUtils.computeMD5Hash(ServiceUtils.fromBase64("your-encryption-customer-key"))));
    request.setUserHeaders(userHeaders);
    ObsObject obsObject=obsClient.getObject(request);
    //可替换读取流的方法
    System.out.println(obsObject.getObjectContent());
}
catch(ObsException e) {
    System.out.println("putObject failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    Map<String, String> headers=e.getResponseHeaders();
    if(headers!=null){
        // 遍历Map的entry打印所有报错相关头域
        for(Map.Entry<String, String> header:headers.entrySet()){
            if(header.getKey().contains("error")){
                System.out.println(header.getKey()+":"+header.getValue());
            }
        }
    }
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("putObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

上传时使用SSE-OBS方式进行服务端加密的对象，下载、获取对象元数据时会自动解密，不用再附加加密相关头域，下载对象见[流式下载](#)，获取对象元数据见[获取对象元数据](#)。

### 📖 说明

- 加密密钥的计算方式，可以参考章节：[如何生成SSE-C方式的加密密钥](#)。

# 20 客户端加密(Java SDK)

## 20.1 客户端加密简介(Java SDK)

客户端加密是指在传输数据前，使用您选择的加密方法与加密信息，在本地进行数据加密，再将加密后结果发送至OBS服务器，同时将加密方法与解密所需的必要辅助信息存储到对象元数据中。在下载时OBS SDK会依据您提供的密钥与存储在对象元数据中的解密辅助信息进行数据解密，直接返回解密后结果。

### 警告

- OBS不会以任何方式存储您的主密钥，您必须自行保证主密钥的正确性与完整性，如果主密钥丢失或用错，将会导致您的加密后数据无法解密，OBS不会承担任何责任。
- OBS SDK会将解密所需的必要辅助信息存储在相应对象的元数据中，在对加密对象进行移动、复制、修改元数据等操作时，您需要保证不会修改辅助信息。如果因您删除/修改解密辅助信息而导致数据无法解密，OBS不承担任何责任。
- 出于数据安全原因，未来版本将会从默认的RSA加密方式修改为指定的加密方式（RSA/ECB/OAEPWITHSHA-256ANDMGF1PADDING），届时会与旧版本不兼容。
- 建议您在密钥信息（master-key-info）中增加sdk版本信息，以防由于安全问题导致的加密算法、密钥长度限制更改导致的不同sdk版本间的数据加解密失败
- 建议您综合安全和性能方面来决定RSA密钥的长度，未来版本会增加密钥长度限制，必须为3072 bits长度以上。
- 建议您综合安全和性能方面来设置secureRandom。
- 建议您使用pkcs8格式的rsa密钥。

## 加密过程与加密套件

OBS Java SDK提供了两个不同的加密套件生成器，分别是基于AES-CTR加密方法的CTRCipherGenerator与基于RSA + AES-CTR加密的CtrRSACipherGenerator。

使用CTRCipherGenerator上传对象时，用户需要提供一个用来加密数据的数据密钥，SDK将会针对每个对象随机生成一个初始值，并使用数据密钥与初始值加密对象，加

密完成后将加密后文件上传至OBS，并将对应的初始值存储至对象元数据中。下载该对象时，用户需要提供对应的数据密钥，SDK会自动获取保存在对象元数据中的初始值，使用数据密钥与初始值解密对象并返回。下载时如果提供了与加密时不同的数据密钥，SDK也将返回不可用的解密后文件。

使用CtrRSACipherGenerator上传对象时，用户需要提供一个RSA 公钥，SDK会针对每个对象随机生成一个数据密钥和初始值，并使用数据密钥与初始值加密对象，加密完成后将加密后文件上传至OBS，随后会使用RSA密钥加密数据密钥，并将加密后的数据密钥与初始值存储至对象元数据中。下载该对象时，用户需要提供对应的RSA私钥，SDK会自动获取保存在对象元数据中的数据密钥与初始值，并使用私钥解密数据密钥，如果提供的私钥与上传时使用的密钥不匹配，将会抛出异常。解密成功后，SDK将会使用解密后的数据密钥与初始值解密对象并返回。

## 加密客户端与普通客户端的接口变更

CryptoObsClient继承自ObsClient，除下表所列举的接口外，其他接口均与ObsClient保持一致。

表 20-1 CryptoObsClient

接口名	CryptoObsClient接口行为
putObject	将流或文件加密后上传至 OBS
getObject	返回解密过的数据流

## 元数据中的解密信息

SDK会将解密所需的必要信息存至对象的对象自定义元数据，并且不会有其他备份，如果您对这些数据进行了修改，将会造成数据无法解密，需要保存的信息及其含义如下表：

表 20-2

参数名称	是否可选	描述
encrypted-algorithm	必选	加密套件的信息。
encrypted-object-key	必选（使用RSA加密套件时）	使用RSA密钥加密后的数据密钥。
encrypted-start	必选	加密该对象所使用的初始值经Base64编码后的字符串。
master-key-info	可选	加密密钥的信息。
plaintext-sha256	可选	加密前对象的sha256，流式上传不计算sha256。
plaintext-content-length	可选	加密前对象的长度，流对象在上传之前无法获取长度。



参数名称	是否可选	描述
encrypted-sha256	可选	加密后对象的sha256，流式上传不计算sha256。

## 20.2 客户端加密接口(Java SDK)

### 初始化 CryptoCipher

OBS Java SDK提供两种加密套件。您可根据使用场景自行选择。

**CtrRSACipherGenerator**继承于**CTRCipherGenerator**，不需要提供数据密钥或初始值，只需要提供RSA公钥或RSA私钥，用以加密和解密随机生成的数据密钥。

**CTRCipherGenerator**仅需提供一个数据密钥，所有对象均使用该数据密钥进行加密。

表 20-3 CtrRSACipherGenerator 类型参数

参数名称	参数类型	是否必选	描述
privateKey	PrivateKey	解密文件时必选（例如 getObject）	<b>参数解释：</b> RSA私钥。 <b>默认取值：</b> 无
publicKey	PublicKey	加密文件时必选（例如 putObject）	<b>参数解释：</b> RSA公钥。 <b>默认取值：</b> 无
masterKeyInfo	String	可选	<b>参数解释：</b> 密钥信息，该信息会存至对象的自定义元数据中，帮助您区分不同密钥，需您自行维护 masterKeyInfo与 密钥的映射关系。 <b>默认取值：</b> 无
secureRandom	SecureRandom	必选	<b>参数解释：</b> 安全随机数生成器，用于随机生成 cryptoKeyBytes和cryptoIvBytes，请您根据业务需求选择设置。 <b>默认取值：</b> 无

参数名称	参数类型	是否必选	描述
needSha256	boolean	可选	<p><b>参数解释:</b> 是否校验加密后数据的sha256, 并设置加密前后的sha256作为自定义元数据。</p> <p><b>说明</b> 为了节省内存开销, SDK 采用了流式计算的方法, 这也就意味着, 普通上传时需要读取并加密文件两次; 在另外由于断点续传上传接口为分段上传接口的封装, 在断点续传下, 则需要读取并加密文件三次。</p> <p><b>取值范围:</b> true: 设置need_sha256为true时, SDK会自动计算待上传对象的加密前sha256值与加密后的sha256值, 并存至对象自定义元数据, 同时也会在发送请求时将加密后的文件的sha256值置于请求头, 服务端收到请求后会计算收到对象的sha256, 如果sha256不一致会返回错误信息。 false: 不计算校验加密后数据的sha256。</p> <p><b>默认取值:</b> false</p>

表 20-4 CTRCipherGenerator 类型参数

参数名称	参数类型	是否必选	描述
masterKeyInfo	String	可选	<p><b>参数解释:</b> 密钥信息, 该信息会存至对象的自定义元数据中, 帮助您区分不同cryptoKeyBytes, 需您自行维护masterKeyInfo与cryptoKeyBytes的映射关系。</p> <p><b>默认取值:</b> 无</p>
cryptoKeyBytes	byte[]	必选	<p><b>参数解释:</b> 加密数据所使用的数据密钥。</p> <p><b>约束限制:</b> 长度必须为32 bytes。</p> <p><b>默认取值:</b> 无</p>

参数名称	参数类型	是否必选	描述
cryptolVBytes	byte[]	可选	<p><b>参数解释:</b> 加密数据时所使用的初始值。</p> <p><b>约束限制:</b></p> <ul style="list-style-type: none"> <li>长度必须为16 bytes。</li> <li>指定时，所有对象均使用该初始值加密；未指定时，SDK会为每个对象随机生成不同初始值。</li> </ul> <p><b>默认取值:</b> 无</p>
secureRandom	SecureRandom	必选	<p><b>参数解释:</b> 安全随机数生成器，用于在未设置cryptoKeyBytes或cryptolVBytes时随机生成缺失的对应参数，请您根据业务需求设置。</p> <p><b>默认取值:</b> 无</p>
needSha256	boolean	可选	<p><b>参数解释:</b> 是否校验加密后数据的sha256，并设置加密前后的sha256作为自定义元数据。</p> <p><b>说明</b> 为了节省内存开销，SDK 采用了流式计算的方法，这也就意味着，普通上传时需要读取并加密文件两次；在另外由于断点续传上传接口为分段上传接口的封装，在断点续传下，则需要读取并加密文件三次。</p> <p><b>取值范围:</b> true：设置need_sha256为true时，SDK会自动计算待上传对象的加密前sha256值与加密后的sha256值，并存至对象自定义元数据，同时也会在发送请求时将加密后的文件的sha256值置于请求头，服务端收到请求后会计算收到对象的sha256，如果sha256不一致会返回错误信息。 false：不计算校验加密后数据的sha256。</p> <p><b>默认取值:</b> false</p>

## 初始化 CryptoObsClient

初始化CryptoObsClient继承自普通ObsClient，详细配置可参考[创建并配置客户端\(Java SDK\)](#)。

## 方法定义

CryptoObsClient(String accessKey, String secretKey, String endPoint,  
CTRCipherGenerator ctrCipherGenerator)

表 20-5 CryptoObsClient 初始化参数

参数名称	描述	建议值
accessKey	<b>参数解释:</b> 访问密钥中的AK。 <b>默认取值:</b> 空字符串, 表示匿名用户	N/A
secretKey	<b>参数解释:</b> 访问密钥中的SK。 <b>默认取值:</b> 空字符串, 表示匿名用户	N/A
endPoint	<b>参数解释:</b> 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)。 <b>默认取值:</b> 无	N/A
ctrCipherGenerator	<b>参数解释:</b> 该加密客户端所使用的 加密套件。 <b>取值范围:</b> <ul style="list-style-type: none"><li>• <a href="#">CtrRSACipherGenerator</a></li><li>• <a href="#">CTRCipherGenerator</a></li></ul> <b>默认取值:</b> 无	N/A

## 支持加密的接口

表 20-6 目前支持加密的接口列表

方法名称	说明
putObject	上传对象
getObject	下载对象

## 示例代码

这是CtrRSACipherGenerator的示例代码：

```
import com.obs.services.ObsConfiguration;
import com.obs.services.crypto.CTRCipherGenerator;
import com.obs.services.crypto.CryptoObsClient;
import com.obs.services.crypto.CtrRSACipherGenerator;
import com.obs.services.exception.ObsException;
import com.obs.services.model.GetObjectRequest;
import com.obs.services.model.ObsObject;
import com.obs.services.model.PutObjectResult;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.spec.InvalidKeySpecException;

public class CtrRSACipherGeneratorDemo001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
        String sk = System.getenv("SECRET_ACCESS_KEY_ID");
        // 【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/SecurityToken，也可以使用其他外部引入方式传入。
        // String securityToken = System.getenv("SECURITY_TOKEN");
        // endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
        String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
        // 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
        //String endPoint = System.getenv("ENDPOINT");
        CtrRSACipherGenerator ctrRSACipherGenerator = null;
        try {
            String examplePrivateKeyPath = "yourRSAPrivateKeyPath";
            String examplePublicKeyPath = "yourRSAPublicKeyPath";
            ObsConfiguration config = new ObsConfiguration();
            PrivateKey privateKeyObj = CtrRSACipherGenerator.importPKCS8PrivateKey(examplePrivateKeyPath);
            PublicKey publicKeyObj = CtrRSACipherGenerator.importPublicKey(examplePublicKeyPath);
            ctrRSACipherGenerator =
                new CtrRSACipherGenerator(
                    "example_master_key_info", true, config.getSecureRandom(), privateKeyObj,
                    publicKeyObj);
        } catch (IllegalArgumentException | IOException | NoSuchAlgorithmException |
            InvalidKeySpecException e) {
            e.printStackTrace();
        }
        assert ctrRSACipherGenerator != null;
        // 创建ObsClient实例
        try (CryptoObsClient cryptoObsClient = new CryptoObsClient(ak, sk, securityToken, endPoint,
            ctrRSACipherGenerator)) {
            String exampleBucketName = "example-bucket";
            String exampleObjectKey = "exampleObjectKey";
            String examplePlainTextFilePath = "examplePlainTextFilePath";
            String exampleDecryptedFilePath = "exampleDecryptedFilePath";
            PutObjectResult putObjectResult =
                cryptoObsClient.putObject(exampleBucketName, exampleObjectKey, new
                File(examplePlainTextFilePath));
            System.out.println("HTTP Code: " + putObjectResult.getStatusCode());
            System.out.println("Etag: " + putObjectResult.getEtag());
            // 客户端加密上传成功
            System.out.println("CtrRSACipherGeneratorDemo001 putObject successfully");

            GetObjectRequest getObjectRequest = new GetObjectRequest(exampleBucketName,
```

```
exampleObjectKey);
    ObsObject obsObject = cryptoObsClient.getObject(getObjectRequest);
    InputStream input = obsObject.getObjectContent();
    byte[] b = new byte[1024];
    FileOutputStream fileOutputStream = new FileOutputStream(exampleDecryptedFilePath);
    int len;
    while ((len = input.read(b)) != -1) {
        fileOutputStream.write(b, 0, len);
    }
    fileOutputStream.close();
    input.close();

    System.out.println("HTTP Code: " + obsObject.getMetadata().getStatusCode());
    // 客户端解密下载成功
    System.out.println("CtrRSACipherGeneratorDemo001 getObject successfully");

    // 验证加密之前的文件和解密之后的文件是否一致
    byte[] plainTextFileSha256 = CTRCipherGenerator.getFileSha256Bytes(examplePlainTextFilePath);
    byte[] decryptedFileSha256 = CTRCipherGenerator.getFileSha256Bytes(exampleDecryptedFilePath);
    String plainTextFileSha256Base64Encoded =
    CTRCipherGenerator.getBase64Info(plainTextFileSha256);
    String decryptedFileSha256Base64Encoded =
    CTRCipherGenerator.getBase64Info(decryptedFileSha256);
    System.out.println("plainTextFileSha256 base64 encoded: " + plainTextFileSha256Base64Encoded);
    System.out.println("decryptedFileSha256 base64 encoded: " + decryptedFileSha256Base64Encoded);
    System.out.println(
        "plainTextFileSha256 equals decryptedFileSha256 ? "
        + decryptedFileSha256Base64Encoded.equals(plainTextFileSha256Base64Encoded));
    System.out.println("CtrRSACipherGeneratorDemo001 successfully");
} catch (ObsException e) {
    System.out.println("CtrRSACipherGeneratorDemo001 failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code: " + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message: " + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
} catch (Exception e) {
    System.out.println("CtrRSACipherGeneratorDemo001 putObject failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

这是CTRCipherGenerator的示例代码：

```
import com.obs.services.crypto.CTRCipherGenerator;
import com.obs.services.crypto.CryptoObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.GetObjectRequest;
import com.obs.services.model.ObsObject;
import com.obs.services.model.PutObjectResult;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;

public class CTRCipherGeneratorDemo001 {
    public static void main(String[] args) {
        // 您可以通过环境变量获取访问密钥AK/SK，也可以使用其他外部引入方式传入。如果使用硬编码可能会存在泄露风险。
        // 您可以登录访问管理控制台获取访问密钥AK/SK
        String ak = System.getenv("ACCESS_KEY_ID");
```

```
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
// 【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/SecurityToken，也可以使用其他外部引入方式传入。
// String securityToken = System.getenv("SECURITY_TOKEN");
// endPoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
//String endPoint = System.getenv("ENDPOINT");
CTRCipherGenerator ctrCipherGenerator = null;
try {
    SecureRandom secureRandom = SecureRandom.getInstanceStrong();
    // 注意 Linux 下SecureRandom.getInstanceStrong()在熵不够的情况下可能阻塞，建议同时增加相关的
    // 补熵机制，或者以其他方式设置SecureRandom
    byte[] exampleMasterKey = new byte[CTRCipherGenerator.CRYPTO_KEY_BYTES_LEN];
    secureRandom.nextBytes(exampleMasterKey);
    // 这里的exampleMasterKey用随机生成的主密钥演示，使用时请您保证密钥长度为32,并妥善保管密
    // 钥。
    ctrCipherGenerator =
        new CTRCipherGenerator("example_master_key_info", exampleMasterKey, true,
secureRandom);
} catch (IllegalArgumentException | NoSuchAlgorithmException e) {
    e.printStackTrace();
}
assert ctrCipherGenerator != null;
// 创建ObsClient实例
try (CryptoObsClient cryptoObsClient =
    new CryptoObsClient(ak, sk, securityToken, endPoint, ctrCipherGenerator)) {
    String exampleBucketName = "example-bucket";
    String exampleObjectKey = "exampleObjectKey";
    String examplePlainTextFilePath = "examplePlainTextFilePath";
    String exampleDecryptedFilePath = "exampleDecryptedFilePath";
    PutObjectResult putObjectResult =
        cryptoObsClient.putObject(exampleBucketName, exampleObjectKey, new
File(examplePlainTextFilePath));
    System.out.println("HTTP Code: " + putObjectResult.getStatusCode());
    System.out.println("Etag: " + putObjectResult.getEtag());
    // 客户端加密上传成功
    System.out.println("CTRCipherGeneratorDemo001 putObject successfully");

    GetObjectRequest getObjectRequest = new GetObjectRequest(exampleBucketName,
exampleObjectKey);
    ObsObject obsObject = cryptoObsClient.getObject(getObjectRequest);
    InputStream input = obsObject.getObjectContent();
    byte[] b = new byte[1024];
    FileOutputStream outputStream = new FileOutputStream(exampleDecryptedFilePath);
    int len;
    while ((len = input.read(b)) != -1) {
        outputStream.write(b, 0, len);
    }
    outputStream.close();
    input.close();

    System.out.println("HTTP Code: " + obsObject.getMetadata().getStatusCode());
    // 客户端解密下载成功
    System.out.println("CTRCipherGeneratorDemo001 getObject successfully");

    // 验证加密之前的文件和解密之后的文件是否一致
    byte[] plainTextFileSha256 = CTRCipherGenerator.getFileSha256Bytes(examplePlainTextFilePath);
    byte[] decryptedFileSha256 = CTRCipherGenerator.getFileSha256Bytes(exampleDecryptedFilePath);
    String plainTextFileSha256Base64Encoded =
CTRCipherGenerator.getBase64Info(plainTextFileSha256);
    String decryptedFileSha256Base64Encoded =
CTRCipherGenerator.getBase64Info(decryptedFileSha256);
    System.out.println("plainTextFileSha256 base64 encoded: " + plainTextFileSha256Base64Encoded);
    System.out.println("decryptedFileSha256 base64 encoded: " + decryptedFileSha256Base64Encoded);
    System.out.println(
        "plainTextFileSha256 equals decryptedFileSha256 ? "
        + decryptedFileSha256Base64Encoded.equals(plainTextFileSha256Base64Encoded));
```

```
        System.out.println("CTRCipherGeneratorDemo001 successfully");
    } catch (ObsException e) {
        System.out.println("CTRCipherGeneratorDemo001 failed");
        // 请求失败,打印http状态码
        System.out.println("HTTP Code: " + e.getResponseCode());
        // 请求失败,打印服务端错误码
        System.out.println("Error Code:" + e.getErrorCode());
        // 请求失败,打印详细错误信息
        System.out.println("Error Message: " + e.getErrorMessage());
        // 请求失败,打印请求id
        System.out.println("Request ID:" + e.getErrorRequestId());
        System.out.println("Host ID:" + e.getErrorHostId());
    } catch (Exception e) {
        System.out.println("CTRCipherGeneratorDemo001 putObject failed");
        // 其他异常信息打印
        e.printStackTrace();
    }
}
```



# 21 问题定位(Java SDK)

## 21.1 问题定位方法(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

使用OBS Java SDK对接OBS服务可能会遇到许多问题，您可以通过下面介绍的步骤进行问题分析和定位：

- 步骤1** 确保使用的是OBS Java SDK的最新版本，您可以从[这里](#)下载最新版本；
- 步骤2** 确保开启了OBS Java SDK日志功能，开启方式参见[日志分析](#)章节，通常建议的日志级别为WARN；
- 步骤3** 确保使用OBS Java SDK的程序代码遵照[OBS客户端通用示例](#)编写，所有ObsClient的接口调用都进行了异常处理，例如上传对象的示例如下：

```
ObsClient obsClient = null;
try
{
    String endPoint = "https://your-endpoint";
    // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca_01_0003.html
    String ak = System.getenv("ACCESS_KEY_ID");
    String sk = System.getenv("SECRET_ACCESS_KEY_ID");

    obsClient = new ObsClient(ak, sk, endPoint);
    HeaderResponse response = obsClient.putObject("bucketname", "objectname", new
    ByteArrayInputStream("Hello OBS".getBytes()));
    // 可选：调用成功后，记录调用成功的HTTP状态码和服务端请求ID
    System.out.println(response.getStatusCode());
    System.out.println(response.getRequestId());
}
catch (ObsException e)
{
    // 推荐：发生异常后，记录失败的HTTP状态码、服务端错误码、服务端请求ID等
    System.out.println("HTTP Code: " + e.getResponseCode());
}
```

```
System.out.println("Error Code:" + e.getErrorCode());  
System.out.println("Request ID:" + e.getErrorRequestId());  
// 推荐：发生异常后，记录异常堆栈信息  
e.printStackTrace(System.out);  
}
```

### 📖 说明

您可以从[这里](#)查看关于ObsException的详细说明。

- 步骤4** 当调用ObsClient的接口发生异常时，从ObsException异常或日志文件中获取**HTTP状态码、OBS服务端错误码**后进行对照，排查异常原因；
- 步骤5** 如果通过步骤4未能排查到异常原因，可从ObsException异常或日志文件中获取OBS服务端请求ID后联系OBS服务端运维团队定位异常原因；
- 步骤6** 如果从ObsException异常或日志文件中无法获取OBS服务端请求ID，请收集ObsException的异常堆栈信息并联系OBS客户端运维团队定位异常原因。

----结束

## 21.2 依赖缺失和依赖冲突的解决(Java SDK)

依赖缺失和依赖冲突是 Java 开发中的常见问题，在集成 SDK 的过程中也会遇到，在应用编译和运行时报错 ClassNotFoundException 与 NoClassDefFoundError 时可怀疑是否是依赖问题而导致，针对不同情况参照下述步骤进行排查和解决。

### 依赖缺失

通过 maven 等包管理插件引入 OBS SDK 时，包管理工具会自动下载相关依赖，其他情况下需要您自行下载依赖包并添加至工程，最新版 SDK 依赖的三方组件与版本如下：

依赖库名称	版本号	作用
okhttp	4.9.1	发送HTTP请求的组件
okio	2.7.0	okhttp的配套组件
java-xmlbuilder	1.3	构建和解析XML的组件
jackson-core	2.12.5	构建和解析JSON的组件
jackson-databind	2.12.5	jackson-core的配套组件
jackson-annotations	2.12.5	jackson-core的配套组件

其他版本的依赖三方库可参考 Maven 中心仓库中 SDK 的 Compile Dependencies 说明。

### 依赖冲突

当您项目中存在多个版本的 OBS Java SDK 软件包，或多个版本的第三方依赖库时，有可能会产生依赖冲突问题。当存在旧版本 SDK 时，建议您删除旧版本软件包，升级至新版；当存在多个版本的第三方依赖库时，请将产生冲突的三方依赖替换为 SDK 的指定版本。

## 使用 Bundle 包

当您的工程存在多种三方依赖包，或多个版本的三方依赖包而又无法删除时，可使用 Bundle 版 SDK。该 Bundle 包中打包了除了 log4j 以外的所有三方依赖，可以不依赖其他三方软件包直接使用，相应的，Bundle 包大小也比普通 SDK 更大，在空间敏感的场景下请评估后使用。Bundle 包的 maven 配置如下：

```
<dependency>
  <groupId>com.huaweicloud</groupId>
  <artifactId>esdk-obs-java-bundle</artifactId>
  <version>[3.21.11,)</version>
</dependency>
```

## 21.3 其他高频问题汇总(Java SDK)

### 须知

开发过程中，您有任何问题可以在github上[提交issue](#)，或者在[华为云对象存储服务论坛](#)中发帖求助。

### SignatureDoesNotMatch 签名不匹配

HTTP Code: 403  
Error Code: SignatureDoesNotMatch

此类错误一般有三种原因：

1. 初始化ObsClient时传入的SK有误，解决方法：检查SK，确保正确；
2. 旧版本OBS Java SDK的BUG，解决方法：升级SDK到最新版本；
3. OBS Java SDK 2.1.x版本与其依赖库Apache HttpClient的兼容性问题，解决方法：使用固定版本的依赖库，httpcore-4.4.4和httpClient-4.5.3。

### MethodNotAllowed 方法不允许

HTTP Code: 405  
Error Code: MethodNotAllowed

此错误一般是请求的OBS服务端未上线ObsClient接口依赖的特性，请联系OBS运维团队进行进一步确认。

### BucketAlreadyOwnedByYou 创桶失败

HTTP Code: 409  
Error Code: BucketAlreadyOwnedByYou

OBS中的桶要求全局唯一，即使在不同区域，也不允许出现同名桶。解决方法：如果调用ObsClient.createBucket接口出现该异常，请确认该桶是否已存在。您可以通过如下两种方式判断该桶是否已存在。

方式一（推荐）：调用ObsClient.listBuckets接口查询您拥有的全部桶列表后判断该桶是否已经存在；

方式二：调用ObsClient.headBucket接口判断该桶是否已经存在。

## 📖 说明

ObsClient.headBucket接口只能访问到当前区域下的桶，而ObsClient.listBuckets接口能访问到所有区域下的桶。

## BucketAlreadyExists 创桶失败

```
HTTP Code: 409
Error Code: BucketAlreadyExists
```

OBS中的桶要求全局唯一，即使在不同区域，也不允许出现同名桶。解决方法：如果调用ObsClient.createBucket接口出现该异常，表明其他用户已创建了该桶，请换一个桶名后重新创建。

## 连接超时

```
HTTP Code: 408
Caused by: java.net.ConnectException: Connection timed out: connect
    at java.net.DualStackPlainSocketImpl.waitForConnect(Native Method)
    at java.net.DualStackPlainSocketImpl.socketConnect(DualStackPlainSocketImpl.java:85)
```

此类错误一般有三种原因：

1. 初始化ObsClient时传入的Endpoint有误，解决方法：检查Endpoint，确保正确；
2. 客户端到OBS服务端的网络异常，导致无法访问，解决方法：检查客户端到OBS服务端的网络健康状况；
3. DNS解析出的OBS服务域名无法访问，解决方法：联系OBS运维团队。

## 读写超时

```
HTTP Code: 408
Error Code: RequestTimeOut
Caused by: java.net.SocketTimeoutException: timeout
    at okio.Okio$4.newTimeoutException(Okio.java:232)
    at okio.AsyncTimeout.exit(AsyncTimeout.java:285)
    at okio.AsyncTimeout$2.read(AsyncTimeout.java:241)
```

此类错误一般有两种原因：

1. 客户端到OBS服务端的网络时延过大，解决方法：检查客户端到OBS服务端的网络健康状况；
2. 客户端到OBS服务端的网络异常，导致无法访问，解决方法：检查客户端到OBS服务端的网络健康状况。

## 异常返回值为-1

```
HTTP Code: -1
```

此类错误一般有三种原因：

1. 使用了旧版的OBS Java SDK并且发生了连接超时或读写超时的异常，解决方法：参见[连接超时](#)和[读写超时](#)的解决方法；
2. 旧版OBS Java SDK的BUG，解决方法：升级到最新版本的SDK，可以从[这里](#)下载最新版本；
3. 服务端返回了异常的结果，导致SDK解析返回结果时出现了不可预期的错误，解决方法：尝试从日志中获取OBS服务端请求ID，并联系OBS运维团队。

## ObsException 中无法获取错误码

此类错误一般有两种原因：

1. 调用ObsClient.getBucketMetadata或ObsClient.getObjectMetadata报错，此种场景下由于后台使用的是HEAD请求，服务端不会返回错误码，解决方法：使用ObsException.getResponseCode获取HTTP状态码，根据状态码分析可能原因，如403一般代表无权限访问，404一般代表桶或对象不存在；如无法定位原因，可从ObsException中获取OBS服务端请求ID后联系OBS运维团队；
2. 初始化ObsClient时传入的Endpoint通过DNS解析后的IP不是有效的OBS服务端，解决方法：检查Endpoint配置是否正确，如Endpoint确认无误，联系OBS运维团队。

## UnknownHostException 域名无法解析异常

```
Caused by: java.net.UnknownHostException: bucketname.unknowndomain.com
    at java.net.Inet6AddressImpl.lookupAllHostAddr(Native Method)
    at java.net.InetAddress$1.lookupAllHostAddr(InetAddress.java:901)
    at java.net.InetAddress.getAddressesFromNameService(InetAddress.java:1293)
```

此类错误一般有两种原因：

1. 初始化ObsClient时传入的Endpoint有误，解决方法：检查Endpoint，确保正确；
2. DNS无法解析OBS服务域名，解决方法：联系OBS运维团队。

## NullPointerException 空指针异常

```
Exception in thread "main" java.lang.NullPointerException
    at com.obs.services.internal.RestStorageService.isCName(RestStorageService.java:1213)
    at com.obs.services.ObsClient.doActionWithResult(ObsClient.java:2805)
```

此类错误一般有两种原因：

1. 使用ObsClient.close接口关闭ObsClient后，再次调用ObsClient的其他接口，解决方法：仅在应用程序退出前调用ObsClient.close接口释放资源；
2. 旧版OBS Java SDK的BUG，解决方法：升级到最新版本的SDK，可以从[这里](#)下载最新版本。

## 连接泄露问题

```
A connection to xxx was leaked. Did you forget to close a response body?
```

此错误通常是调用ObsClient.getObject接口获取下载对象的数据流后未正常关闭，解决方法：确保在finally语句块中调用了ObsObject.getObjectContent.close方法关闭连接。

## 升级 SDK 后 okhttp 报错问题

```
Exception in thread "main" java.lang.NoSuchMethodError: 'okhttp3.RequestBody
okhttp3.RequestBody.create(java.lang.String, okhttp3.MediaType)'
    at com.obs.services.internal.RestConnectionService.createRequestBuilder(RestConnectionService.java:157)
    at com.obs.services.internal.RestConnectionService.setupConnection(RestConnectionService.java:148)
    at com.obs.services.internal.RestConnectionService.setupConnection(RestConnectionService.java:124)
    at com.obs.services.internal.RestStorageService.performRequest(RestStorageService.java:395)
    at com.obs.services.internal.RestStorageService.performRequest(RestStorageService.java:388)
```

此报错是因为升级SDK后okhttp使用了老版本的原因，请参考[依赖缺失和依赖冲突的解决\(Java SDK\)](#)，升级okhttp到指定版本。

## 升级 SDK 后报错 StackOverflowError 问题

```
Caused by: java.lang.StackOverflowError
  at sun.misc.URLClassPath.getResource(URLClassPath.java:211) ~[?:1.8.0_91]
  at java.net.URLClassLoader$1.run(URLClassLoader.java:365) ~[?:1.8.0_91]
  at java.net.URLClassLoader$1.run(URLClassLoader.java:362) ~[?:1.8.0_91]
  at java.security.AccessController.doPrivileged(Native Method) ~[?:1.8.0_91]
  at java.net.URLClassLoader.findClass(URLClassLoader.java:361) ~[?:1.8.0_91]
  at java.lang.ClassLoader.loadClass(ClassLoader.java:424) ~[?:1.8.0_91]
  at java.lang.ClassLoader.loadClass(ClassLoader.java:357) ~[?:1.8.0_91]
  at org.apache.catalina.loader.WebappClassLoaderBase.loadClass(WebappClassLoaderBase.java:1806)
```

检查jvm参数xss，建议将xss参数设置为1M。xss是jvm启动的每个线程分配的内存大小，默认JDK1.4中是256K，JDK1.5+中是1M。

## 报错 SSL peer shut down incorrectly

```
javax.net.ssl.SSLException: SSL peer shut down incorrectly
  at sun.security.ssl.InputRecord.readV3Record(InputRecord.java:596)
  at sun.security.ssl.InputRecord.read(InputRecord.java:532)
  at sun.security.ssl.SSLSocketImpl.readRecord(SSLSocketImpl.java:975)
  at sun.security.ssl.SSLSocketImpl.readDataRecord(SSLSocketImpl.java:933)
  at sun.security.ssl.AppInInputStream.read(AppInInputStream.java:105)
  at obs.shaded.okio.Okio$2.read(Okio.java:140)
  at obs.shaded.okio.AsyncTimeout$2.read(AsyncTimeout.java:237)
  at obs.shaded.okio.RealBufferedSource.read(RealBufferedSource.java:51)
  at obs.shaded.okhttp3.internal.http1.Http1ExchangeCodec
$AbstractSource.read(Http1ExchangeCodec.java:389)
  at obs.shaded.okhttp3.internal.http1.Http1ExchangeCodec
$FixedLengthSource.read(Http1ExchangeCodec.java:427)
  at obs.shaded.okhttp3.internal.connection.Exchange$ResponseBodySource.read(Exchange.java:286)
  at obs.shaded.okio.RealBufferedSource$1.read(RealBufferedSource.java:447)
  at java.util.zip.InflaterInputStream.fill(InflaterInputStream.java:238)
  at java.util.zip.InflaterInputStream.read(InflaterInputStream.java:158)
  at java.util.zip.GZIPInputStream.read(GZIPInputStream.java:117)
  at sun.nio.cs.StreamDecoder.readBytes(StreamDecoder.java:284)
  at sun.nio.cs.StreamDecoder.implRead(StreamDecoder.java:326)
  at sun.nio.cs.StreamDecoder.read(StreamDecoder.java:178)
  at java.io.InputStreamReader.read(InputStreamReader.java:184)
  at java.io.BufferedReader.fill(BufferedReader.java:161)
  at java.io.BufferedReader.readLine(BufferedReader.java:324)
  at java.io.BufferedReader.readLine(BufferedReader.java:389)
```

客户端下载读取文件流时，请不要按行读取，可参考[流式下载](#)的Demo。

## 其他问题

请参考[FAQ](#)，查看更多问题的解决方法。

# 22 异常处理(Java SDK)

## 22.1 HTTP 状态码(Java SDK)

OBS服务端遵照HTTP规范，在接口调用完成均会返回标准的HTTP状态码，HTTP状态码分类以及OBS中常见的HTTP状态码如下：

- HTTP状态码分类：

分类	分类描述
1XX	信息，服务器收到请求，需要请求者继续执行操作，一般对客户调用函数不可见。
2XX	成功，操作被成功接收并处理。
3XX	重定向，需要进一步的操作以完成请求，一般对客户调用函数不可见。
4XX	客户端错误，请求包含语法错误或无法完成请求。
5XX	服务器错误，服务器在处理请求的过程中发生了错误。

- OBS中常见的HTTP状态码及其含义：

HTTP状态码	描述	常见原因
400 Bad Request	请求参数错误	<ul style="list-style-type: none"><li>• 请求参数不合法；</li><li>• 客户端携带MD5请求后一致性校验失败；</li><li>• 无效的参数（使用SDK时传递了不合法的参数）；</li><li>• 无效的桶名（使用了不合法的桶名）。</li></ul>

HTTP状态码	描述	常见原因
403 Forbidden	拒绝访问	<ul style="list-style-type: none"> <li>客户端请求中携带的签名和服务端计算出的签名不匹配（一般是AK/SK错误）；</li> <li>权限不足（账号对请求的资源无权限）；</li> <li>账号欠费；</li> <li>桶的空间不足（出现在对桶设置了配额场景）；</li> <li>无效的AK；</li> <li>客户端时间和服务端时间相差过大（客户端所在机器的时间与NTP服务不同步）。</li> </ul>
404 Not Found	请求的资源不存在	<ul style="list-style-type: none"> <li>桶不存在；</li> <li>对象不存在；</li> <li>桶的策略配置不存在（桶CORS配置不存在、桶Policy配置不存在等）；</li> <li>分段上传任务不存在。</li> </ul>
405 Method Not Allowed	请求的方法不支持	请求的方法/特性未在该桶所在的区域上线。
408 Request Timeout	请求超时	服务端与客户端Socket连接超时。
409 Conflict	请求冲突	<ul style="list-style-type: none"> <li>在不同区域重复创建同名桶；</li> <li>尝试删除非空桶。</li> </ul>
500 Internal Server Error	服务端内部错误	服务端内部错误。
503 Service Unavailable	服务不可用	服务端暂时不可访问。

## 22.2 OBS 服务端错误码(Java 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：



HTTP状态码	错误码	错误信息	处理措施
301 Moved Permanently	PermanentRedirect	尝试访问的桶必须使用指定的地址，请将以后的请求发送到这个地址。	按照返回的重定向地址发送请求。
301 Moved Permanently	WebsiteRedirect	Website请求缺少bucketName。	携带桶名后重试。
307 Moved Temporarily	TemporaryRedirect	临时重定向，当DNS更新时，请求将被重定向到桶。	会自动重定向，也可以将请求发送到重定向地址。
400 Bad Request	BadDigest	客户端指定的对象内容的MD5值与系统接收到的内容MD5值不一致。	检查头域中携带的MD5与消息体计算出来的MD5是否一致。
400 Bad Request	BadDomainName	域名不合法。	使用合法的域名。
400 Bad Request	BadRequest	请求参数不合法。	根据返回的错误消息体提示进行修改。
400 Bad Request	CustomDomainAlreadyExist	配置了已存在的域。	已经配置过了，不需要再配置。
400 Bad Request	CustomDomainNotExist	删除不存在的域。	未配置或已经删除，无需删除。
400 Bad Request	EntityTooLarge	用户POST上传的对象大小超过了条件允许的最大大小。	修改POST上传的policy中的条件或者减少对象大小。
400 Bad Request	EntityTooSmall	用户POST上传的对象大小小于条件允许的最小大小。	修改POST上传的policy中的条件或者增加对象大小。
400 Bad Request	IllegalLocationConstraintException	用户未带Location在非默认Region创桶。	请求发往默认Region创桶或带非默认Region的Location创桶。
400 Bad Request	IncompleteBody	由于网络原因或其他问题导致请求体未接受完整。	重新上传对象。
400 Bad Request	IncorrectNumberOfFilesInPostRequest	每个POST请求都需要带一个上传的文件。	带上一个上传文件。
400 Bad Request	InvalidArgument	无效的参数。	根据返回的错误消息体提示进行修改。

HTTP状态码	错误码	错误信息	处理措施
400 Bad Request	InvalidBucket	请求访问的桶已不存在。	更换桶名。
400 Bad Request	InvalidBucketName	请求中指定的桶名无效，超长或带不允许的特殊字符。	更换桶名。
400 Bad Request	InvalidEncryptionAlgorithmError	错误的加密算法。下载SSE-C加密的对象，携带的加密头域错误，导致不能解密。	携带正确的加密头域下载对象。
400 Bad Request	InvalidLocationConstraint	创建桶时，指定的Location不合法或不存在。	指定正确的Location创桶。
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。	按照正确的段和entity tag合并段。
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。	按段号升序排列后重新合并。
400 Bad Request	InvalidPolicyDocument	表单中的内容与策略文档中指定的条件不一致。	根据返回的错误消息体提示修改构造表单的policy重试。
400 Bad Request	InvalidRedirectLocation	无效的重定向地址。	指定正确的地址。
400 Bad Request	InvalidRequest	无效请求。	根据返回的错误消息体提示进行修改。
400 Bad Request	InvalidRequestBody	请求体无效，需要消息体的请求没有上传消息体。	按照正确的格式上传消息体。
400 Bad Request	InvalidTargetBucketForLogging	delivery group对目标桶无ACL权限。	对目标桶配置ACL权限后重试。
400 Bad Request	KeyTooLongError	提供的Key过长。	使用较短的Key。
400 Bad Request	KMS.DisabledException	SSE-KMS加密方式下，主密钥被禁用。	更换密钥后重试，或联系技术支持。

HTTP状态码	错误码	错误信息	处理措施
400 Bad Request	KMS.NotFoundException	SSE-KMS加密方式下，主密钥不存在。	携带正确的主密钥重试。
400 Bad Request	MalformedACLError	提供的XML格式错误，或者不符合要求的格式。	使用正确的XML格式重试。
400 Bad Request	MalformedError	请求中携带的XML格式不正确。	使用正确的XML格式重试。
400 Bad Request	MalformedLoggingStatus	Logging的XML格式不正确。	使用正确的XML格式重试。
400 Bad Request	MalformedPolicy	Bucket policy检查不通过。	根据返回的错误消息体提示结合桶policy的要求进行修改。
400 Bad Request	MalformedQuotaError	Quota的XML格式不正确。	使用正确的XML格式重试。
400 Bad Request	MalformedXML	当用户发送了一个配置项的错误格式的XML会出现这样的错误。	使用正确的XML格式重试。
400 Bad Request	MaxMessageLengthExceeded	拷贝对象，带请求消息体。	拷贝对象不带消息体重试。
400 Bad Request	MetadataTooLarge	元数据消息头超过了允许的最大元数据大小。	减少元数据消息头。
400 Bad Request	MissingRegion	请求中缺少Region信息，且系统无默认Region。	请求中携带Region信息。
400 Bad Request	MissingRequestBodyError	当用户发送一个空的XML文档作为请求时会发生。	提供正确的XML文档。
400 Bad Request	MissingRequiredHeader	请求中缺少必要的头域。	提供必要的头域。
400 Bad Request	MissingSecurityHeader	请求缺少一个必须的头。	提供必要的头域。
400 Bad Request	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。	删除部分桶后重试。
400 Bad Request	TooManyCustomDomains	配置了过多的用户域	删除部分用户域后重试。

HTTP状态码	错误码	错误信息	处理措施
400 Bad Request	TooManyWrongSignature	因高频错误请求被拒绝服务。	更换正确的Access Key后重试。
400 Bad Request	UnexpectedContent	该请求需要消息体而客户端没带，或该请求不需要消息体而客户端带了。	根据说明重试。
400 Bad Request	AZRedundancyTypeNotSupported	当前region不支持此AZRedundancy的桶	选择合适的AZ Redundancy创桶
400 Bad Request	UserKeyMustBeSpecified	该操作只有特殊用户可使用。	请联系技术支持。
403 Forbidden	AccessDenied	拒绝访问，请求没有携带日期头域或者头域格式错误。	请求携带正确的日期头域。
403 Forbidden	AccessForbidden	权限不足，桶未配置CORS或者CORS规则不匹配。	修改桶的CORS配置，或者根据桶的CORS配置发送匹配的OPTIONS请求。
403 Forbidden	AllAccessDisabled	用户无权限执行某操作。桶名为禁用关键字。	更换桶名。
403 Forbidden	DeregisterUserId	用户已经注销。	充值或重新开户。
403 Forbidden	InArrearOrInsufficientBalance	用户欠费或余额不足而没有权限进行某种操作。	充值。
403 Forbidden	InsufficientStorageSpace	存储空间不足。	超过配额限制，增加配额或删除部分对象。
403 Forbidden	InvalidAccessKeyId	系统记录中不存在客户提供的Access Key Id。	携带正确的Access Key Id。
403 Forbidden	NotSignedUp	你的账户还没有在系统中注册，必须先系统中注册了才能使用该账户。	先注册OBS服务。
403 Forbidden	RequestTimeTooSkewed	请求的时间与服务器的时间相差太大。	检查客户端时间是否与当前时间相差太大。

HTTP状态码	错误码	错误信息	处理措施
403 Forbidden	SignatureDoesNot Match	请求中带的签名与系统计算得到的签名不一致。	检查你的Secret Access Key和签名计算方法。
403 Forbidden	Unauthorized	用户未实名认证。	请实名认证后重试。
404 Not Found	NoSuchBucket	指定的桶不存在。	先创桶再操作。
404 Not Found	NoSuchBucketPolicy	桶policy不存在。	先配置桶policy。
404 Not Found	NoSuchCORSConfiguration	CORS配置不存在。	先配置CORS。
404 Not Found	NoSuchCustomDomain	请求的用户域不存在。	先设置用户域。
404 Not Found	NoSuchKey	指定的Key不存在。	先上传对象。
404 Not Found	NoSuchLifecycleConfiguration	请求的LifeCycle不存在。	先配置LifeCycle。
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。	使用存在的段或重新初始化段。
404 Not Found	NoSuchVersion	请求中指定的version ID与现存的所有版本都不匹配。	使用正确的version ID。
404 Not Found	NoSuchWebsiteConfiguration	请求的Website不存在。	先配置Website。
405 Method Not Allowed	MethodNotAllowed	指定的方法不允许操作在请求的资源上。 对应返回的Message为： Specified method is not supported.	方法不允许。
408 Request Timeout	RequestTimeout	用户与Server之间的socket连接在超时时间内没有进行读写操作。	检查网络后重试，或联系技术支持。

HTTP状态码	错误码	错误信息	处理措施
409 Conflict	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。	更换桶名。
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。	不需要再创桶了。
409 Conflict	BucketNotEmpty	用户尝试删除的桶不为空。	先删除桶中对象，然后再删桶。
409 Conflict	FsObjectConflict	用户上传或创建文件失败	检查创建文件规则。例如是否在不允许覆盖的场景下覆盖文件、是否在POSIX语义下在文件下上传文件（将文件当作目录）
409 Conflict	InvalidBucketState	无效的桶状态，配置跨Region复制后不允许关闭桶多版本。	不关闭桶的多版本或取消跨Region复制。
409 Conflict	OperationAborted	另外一个冲突的操作当前正作用在这个资源上，请重试。	等待一段时间后重试。
409 Conflict	ServiceNotSupported	请求的方法服务端不支持。	服务端不支持，请联系技术支持。
411 Length Required	MissingContentLength	必须要提供HTTP消息头中的Content-Length字段。	提供Content-Length消息头。
412 Precondition Failed	PreconditionFailed	用户指定的先决条件中至少有一项没有包含。	根据返回消息体中的Condition提示进行修改。
416 Client Requested Range Not Satisfiable	InvalidRange	请求的range不可获得。	携带正确的range重试。
500 Internal Server Error	InternalServerError	系统遇到内部错误，请重试。	请联系技术支持。
501 Not Implemented	ServiceNotImplemented	请求的方法服务端没有实现。	当前不支持，请联系技术支持。

HTTP状态码	错误码	错误信息	处理措施
503 Service Unavailable	ServiceUnavailable	服务器过载或者内部错误异常。	等待一段时间后重试，或联系技术支持。
503 Service Unavailable	SlowDown	请降低请求频率。	请降低请求频率。

## 22.3 SDK 自定义异常(Java SDK)

SDK自定义异常（ObsException）是由ObsClient统一抛出的异常，继承自java.lang.RuntimeException类。通常是OBS服务端错误，包含**OBS错误码**、错误信息等，便于用户定位问题，并做出适当的处理。

ObsException通常包含以下错误信息：

- ObsException.getResponseCode：HTTP状态码。
- ObsException.getErrorCode：OBS服务端错误码。
- ObsException.getErrorMessage：OBS服务端错误描述。
- ObsException.getErrorRequestId：OBS服务端返回的请求ID。
- ObsException.getErrorHostId：请求的服务端ID。
- ObsException.getResponseHeaders：HTTP响应头信息。

## 22.4 SDK 公共响应头(Java SDK)

调用ObsClient类的相关接口成功后，均会返回公共响应头类，即HeaderResponse类实例（或其子类实例），该类包含了HTTP/HTTPS的响应头信息。

处理公共响应头的示例代码如下：

```
// Endpoint以北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca_01_0003.html
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
// 创建ObsClient实例
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
HeaderResponse response = obsClient.createBucket("bucketname");

// 从公共响应头中获取request-id
System.out.println("\t" + response.getRequestId());

obsClient.close();
```

## 22.5 日志分析(Java SDK)

### 日志开启方式

1. 将OBS SDK包中的log4j2.xml文件放到classpath根目录;
2. 或者调用Log4j2Configurator.setLogConfig直接指定log4j2.xml文件的路径。

#### 📖 说明

您可以从OBS Java SDK的发布软件包中获取默认的日志配置文件log4j2.xml，并在该文件基础上进行修改定制。

### 日志路径

OBS Java SDK的日志路径是通过log4j2.xml指定的，默认存放于JDK系统变量user.dir所代表的路径下。通常包含三个日志文件：

文件名	说明
OBS-SDK.interface_north.log	北向日志，OBS Java SDK与用户第三方应用交互的日志记录。
OBS-SDK.interface_south.log	南向日志，OBS Java SDK与OBS服务端交互的日志记录。
OBS-SDK.access.log	OBS客户端运行日志。

### 日志内容格式

SDK日志格式为：日志时间|线程号|日志级别|日志内容。示例如下：

```
#南向日志
2017-08-21 17:40:07 133|main|INFO |HttpClient cost 157 ms to apply http request
2017-08-21 17:40:07 133|main|INFO |Received expected response code: true
2017-08-21 17:40:07 133|main|INFO |expected code(s): [200, 204].

#北向日志
2017-08-21 17:40:06 820|main|INFO |Storage|1|HTTP+XML|ObsClient|2017-08-21 17:40:05|2017-08-21
17:40:06||0|
2017-08-21 17:40:07 136|main|INFO |Storage|1|HTTP+XML|setObjectAcl|2017-08-21 17:40:06|2017-08-21
17:40:07||0|
2017-08-21 17:40:07 137|main|INFO |ObsClient [setObjectAcl] cost 312 ms
```

### 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TRACE日志信息最丰富，ERROR日志信息最少。

具体说明如下：

- OFF: 关闭级别，如果设置为这个级别，日志打印功能将被关闭。
- TRACE: 跟踪级别，如果设置为这个级别，将打印所有日志信息。通常不建议使用。



- **DEBUG**: 调试级别, 如果设置为这个级别, 除了打印INFO级别以上的信息外, 还将打印每次HTTP/HTTPS请求和响应的头信息, 鉴权算法计算出的StringToSign信息等。
- **INFO**: 信息级别, 如果设置为这个级别, 除了打印WARN级别以上的信息外, 还将打印HTTP/HTTPS请求的耗时时间, ObsClient接口的耗时时间等。
- **WARN**: 告警级别, 如果设置为这个级别, 除了打印ERROR级别以上的信息外, 还将打印一些关键事件的信息, 如重试请求超过最大次数等。
- **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。

## 设置方式

以下代码分别为南向日志、北向日志和OBS客户端运行日志设置了不同的日志级别 (日志配置文件的全部内容参见log4j2.xml配置文件):

```
<!-- north log -->
<Logger name="com.obs.services.AbstractClient" level="INFO" additivity="false">
  <AppenderRef ref="NorthInterfaceLogAppender" />
</Logger>

<!-- south log -->
<Logger name="com.obs.services.internal.RestStorageService" level="WARN" additivity="false">
  <AppenderRef ref="SouthInterfaceLogAppender" />
</Logger>

<!-- access log -->
<Logger name="com.obs.log.AccessLogger" level="ERROR" additivity="false">
  <AppenderRef ref="AccessLogAppender" />
</Logger>
```

# 23 常见问题(Java SDK)

## 23.1 如何使对象可以被匿名用户访问? (Java SDK)

如果想让对象被匿名用户访问,可通过以下三步完成。

**步骤1** 参考[设置对象ACL](#)章节,设置对象的访问权限为公共读。

**步骤2** 参考[如何获取对象URL](#)章节,获取对象的URL提供给匿名用户。

**步骤3** 匿名用户通过浏览器打开获取的URL,可访问到该对象。

----结束

## 23.2 SDK 的重试机制是什么? (Java SDK)

SDK通过[创建并配置客户端](#)章节中的maxErrorRetry参数来实行重试,默认重试3次,建议值为0到5次之间。当调用ObsClient的接口发生网络连接异常或者服务端返回5XX错误时,SDK会进行指数退避重试。

### 须知

- ObsClient.putObject接口,当数据源是非FileInputStream的其他InputStream时,由于数据流不能回读,当发生IO异常时,SDK不会进行重试,需要上层应用程序自行重试;
- ObsClient.getObject接口,当请求成功并返回ObsObject对象后,由于此时已不在SDK的处理逻辑范围,当从ObsObject.getObjectContent读取数据过程中,当发生IO异常时,SDK不会进行重试,需要上层应用程序自行重试。

## 23.3 如何获取桶的静态网站访问地址? (Java SDK)

桶配置成静态网站托管模式后,可通过以下方式拼接桶的静态网站访问地址:

`https://桶名.静态网站托管域名`

### 📖 说明

- 各区域对应的静态网站托管域名可以从[这里](#)查看。
- 例如区域为“华北-北京四”的桶名为“testbucket”的桶配置成静态网站托管模式后，该桶的静态网站访问地址为<https://testbucket.obs-website.cn-north-4.myhuaweicloud.com>。

## 23.4 如何获取对象 URL? (Java SDK)

将桶中的对象权限设置为匿名用户读取权限后，可通过对象的URL直接下载该对象。获取对象URL的方式如下：

方式一，接口查询，ObsClient上传对象后会返回PutObjectResult对象，通过调用该对象的getObjectUrl接口可获取上传对象的URL。示例代码如下：

```
// Endpoint以北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca\_01\_0003.html
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
// 创建ObsClient实例
ObsClient obsClient = new ObsClient(ak, sk, endPoint);
// 调用putObject接口上传对象并获取其返回结果
PutObjectResult result = obsClient.putObject("bucketname", "objectname", new File("localfile"));
// 读取该已上传对象的URL
System.out.println("\t" + result.getObjectUrl());
```

方式二，按<https://桶名.域名/文件夹目录层级/对象名>的方式进行拼接。

### 📖 说明

- 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；
- 各区域对应的域名可以从[这里的](#)终端节点查看。
- 例如需访问区域为“华北-北京四”的桶名为“testbucket”中“test”文件夹下对象名为“test.txt”的对象，则该对象的URL为<https://testbucket.obs.cn-north-4.myhuaweicloud.com/test/test.txt>。

## 23.5 公网环境下如何提高上传大文件速度? (Java SDK)

在公网环境下对于超过100MB的大文件建议通过分段上传方式上传。分段上传是将单个对象拆分为一系列段分别上传。每个段都是对象数据的连续部分。您可以按照任意顺序上传段。如果其中某个段传输失败，可以重新传输该段且不会影响其他段。通过多线程并发上传同一对象的多个段，可大大提高传输效率。

具体代码样例可参见[分段上传](#)。

## 23.6 如何停止正在上传中的任务? (Java SDK)

SDK不支持该特性，需要进行二次开发。可通过中断数据流并捕获异常的方法停止正在上传中的任务。

## 23.7 如何进行分段上传? (Java SDK)

在上传文件时，您可以指定上传文件的某一段的数据来进行分段上传，步骤如下：

- 步骤1** 您需要以AK、SK、Endpoint先初始化一个客户端ObsClient；
- 步骤2** 指定桶名和对象名初始化一个InitiateMultipartUploadRequest，您可以通过InitiateMultipartUploadRequest.setMetadata接口设置将要上传的对象的元数据信息，然后通过ObsClient.initiateMultipartUpload接口初始化一个分段上传任务，这个接口会返回一个全局唯一标示（Upload ID），用于标识本次分段上传任务；
- 步骤3** 指定桶名和对象名初始化一个UploadPartRequest请求，通过UploadPartRequest.setUploadId接口设置该上传段所属的uploadId，setPartNumber接口设置该上传段的段号，setFile接口设置该上传段所属的将要上传的大文件，setPartSize接口设置该上传段的段大小；然后通过ObsClient.uploadPart接口上传该段，这个接口会返回该上传段的ETag值；
- 步骤4** 在所有分段上传完成后，指定桶名、对象名、uploadId、partEtags来初始化一个CompleteMultipartUploadRequest请求，然后通过ObsClient.completeMultipartUpload接口进行合并段操作。

----结束

详细操作可参见[分段上传](#)。

## 23.8 如何进行分段下载? (Java SDK)

在下载对象时，您可以指定下载对象的某一范围内的数据进行分段下载，步骤如下：

- 步骤1** 您需要以AK、SK、Endpoint先初始化一个客户端ObsClient；
- 步骤2** 指定桶名和对象名初始化一个GetObjectRequest请求，您可以通过GetObjectRequest.setRangeStart和GetObjectRequest.setRangeEnd接口设置将要下载的对象数据的开始和结束范围；
- 步骤3** 通过ObsClient.getObject接口发送步骤2的GetObjectRequest请求来进行分段下载。

----结束

详细操作可参见[范围下载](#)。

## 23.9 如果桶内已存在同名对象，如何判定覆盖上传成功? (Java SDK)

上传完成后，您可以调用ObsClient.getObjectMetadata接口获取目标对象大小和最后修改时间，再与数据源进行比较，如果两者大小一致且目标对象的最后修改时间晚于数据源的最后修改时间则表明上传成功，否则上传失败。

ObsClient.getObjectMetadata接口的使用可参见[18.4-获取对象元数据](#)。

## 23.10 对于加密类型的对象如何使用 URL 进行下载? (Java SDK)

如果是SSE-KMS方式进行加密的对象，使用URL下载时服务端会自动解密；如果对象是SSE-C方式进行加密的对象，由于解密需要设置请求头域，所以不能通过浏览器直接访问该对象，如果您是在编写代码来访问加密类型的对象，只需要为请求设置相应的头域信息即可，可参见[代码示例：下载SSE-C加密类型的对象](#)。

### 📖 说明

访问服务端加密对象必须使用HTTPS协议。

## 23.11 如何生成 SSE-C 方式的加密密钥(Java SDK)

以下代码展示了如何生成SSE-C方式的加密密钥，以及密钥的MD5值如何生成：

```
import java.nio.charset.Charset;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class SseCTool {
    public static void main(String[] args) {
        // 加密密钥示例：该密钥可根据业务需要自行修改，但长度必须为32个字符，即256 bit的字符串
        String keyString = System.getenv("ACCESS_KEY_ID");
        // 计算x-obs-server-side-encryption-customer-key头域
        String customerKey = toBase64String(keyString.getBytes(Charset.forName("UTF-8")));
        System.out.println("customer key is : " + customerKey);

        // 计算x-obs-server-side-encryption-customer-key-MD5头域
        String customerKeyMD5 = computeMD5(fromBase64(customerKey));
        System.out.println("md5 for customer key is : " + customerKeyMD5);
    }

    // 计算Base64字符串
    public static String toBase64String(byte[] data) {
        java.util.Base64.Encoder encoder = java.util.Base64.getEncoder();
        return new String(encoder.encode(data), Charset.forName("UTF-8"));
    }

    // 计算MD5字符串
    public static String computeMD5(byte[] b64Data) {
        try {
            MessageDigest md5 = MessageDigest.getInstance("MD5");
            md5.update(b64Data);

            byte[] byteArray = md5.digest();

            return toBase64String(byteArray);
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
            return "";
        }
    }

    public static byte[] fromBase64(String b64Data) {
        java.util.Base64.Decoder decoder = java.util.Base64.getDecoder();
        return decoder.decode(b64Data.getBytes(Charset.forName("UTF-8")));
    }
}
```

## 23.12 如何获取 SecurityToken? (Java SDK)

获取SecurityToken可以参考[IAM开发指导文档](#)示例，也可以通过OBS API提供的示例程序获取，请参考[通过委托获取临时访问密钥和securitytoken](#)。

## 23.13 SDK 是否支持批量上传、下载或复制对象? (Java SDK)

不支持。

目前SDK暂未提供此类接口，您需要自己封装批量上传、下载或复制对象的业务代码。步骤如下：

**步骤1** 列举出所有待上传、下载或者复制的对象。可参考[列举对象](#)章节，列举待下载的对象。

**步骤2** 对列举出的对象调用单个对象的上传（上传对象）、下载（下载对象）或复制（复制对象）接口。

----**结束**

以批量上传对象为例，示例代码如下：

```
// Endpoint以北京四为例，其他地区请按实际情况填写。
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；本示例以ak和sk保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/usermanual-ca/ca_01_0003.html
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
final String bucketName = "bucketname";
// 定义桶内对象的前缀
final String objectPre = "object/";
// 待上传的文件夹
final String localDirPath = "localDirPath";
final List<File> list = new ArrayList<>();
// 扫描文件夹下所有对象
static void listFiles(File file){
    File[] fs = file.listFiles();
    assert fs != null;
    if (fs.length < 1){
        // 如果是空文件夹也需要上传，将其添加到列表中
        list.add(file);
    }else{
        for (File f:fs){
            if (f.isDirectory()){
                listFiles(f);
            }
            if (f.isFile()){
                // 添加待上传对象到列表中
                list.add(f);
            }
        }
    }
}
// 遍历待上传的文件夹，获取所有待上传对象
File file = new File(localDirPath);
listFiles(file);
```

```
// 创建ObsClient实例
final ObsClient obsClient = new ObsClient(ak, sk, endPoint);

// 初始化线程池
ExecutorService executorService = Executors.newFixedThreadPool(20);

// 执行并发上传
for (File f:list){
    executorService.execute(() -> {
        if (f.isDirectory()){
            // 如果是空文件夹, 则在桶内创建对应的空文件夹对象
            String remoteObjectKey = objectPre + f.getPath().substring(localDirPath.length() + 1) + "/";
            obsClient.putObject(bucketName, remoteObjectKey, new ByteArrayInputStream(new byte[0]));
        }else{
            String remoteObjectKey = objectPre + f.getPath().substring(localDirPath.length() + 1);
            obsClient.putObject(bucketName, remoteObjectKey, new File(f.getPath()));
        }
    });
}

// 等待上传完成
executorService.shutdown();
while (!executorService.isTerminated())
{
    try
    {
        {
            executorService.awaitTermination(5, TimeUnit.SECONDS);
        }
        catch (InterruptedException e)
        {
            e.printStackTrace();
        }
    }
}
// 关闭obsClient
try {
    obsClient.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

#### 说明

您可以使用多线程并发执行上传/下载/复制操作, 以提高效率。

## 23.14 如何指定 Content-SHA256?(Java 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), 上传对象示例代码如下:

```
import com.obs.services.ObsClient;
import com.obs.services.exception.ObsException;
import com.obs.services.model.PutObjectRequest;

import java.io.File;
import java.io.InputStream;
import java.nio.file.Files;
import java.security.MessageDigest;

public class PutObjectWithSha256 {
    public static void main(String[] args) {
        // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险, 建议在配置文件或者环境变量中密文存放, 使用时解密, 确保安全; 本示例以ak和sk保存在环境变量中为例, 运行本示例前请先在本地环境中设置环境变量ACCESS_KEY_ID和SECRET_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/
    }
}
```

```
usermanual-ca/ca_01_0003.html
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY_ID");
// 【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/SecurityToken，也可以使用其他外部引入方式传入。
String securityToken = System.getenv("SECURITY_TOKEN");

// endpoint填写桶所在的endpoint，此处以华北-北京四为例，其他地区请按实际情况填写。
// 您可以通过环境变量获取endPoint，也可以使用其他外部引入方式传入。
// String endPoint = System.getenv("ENDPOINT");
String endPoint = "https://obs.cn-north-4.myhuaweicloud.com";

// 创建ObsClient实例
try (ObsClient obsClient = new ObsClient(ak, sk, securityToken, endPoint)) {
    // 本示例用于文件上传到examplebucket桶中的exampleObjectKey,并指定content-sha256头域。

    String bucketName = "examplebucket";
    String objectKey = "exampleObjectKey";
    File localFile = new File("localFilePath");
    PutObjectRequest putObjectRequest = new PutObjectRequest();
    putObjectRequest.setBucketName(bucketName);
    putObjectRequest.setObjectKey(objectKey);
    putObjectRequest.setFile(localFile);
    // 计算文件sha256
    InputStream fis = Files.newInputStream(localFile.toPath());
    byte[] buffer = new byte[1024];
    MessageDigest sha256 = MessageDigest.getInstance("SHA-256");
    for (int numRead = 0; (numRead = fis.read(buffer)) > 0; ) {
        sha256.update(buffer, 0, numRead);
    }
    fis.close();
    // 转为16进制
    StringBuilder sha256Builder = new StringBuilder();
    for (byte aB : sha256.digest()) {
        sha256Builder.append(String.format("%02x", aB));
    }
    // 添加文件sha256到自定义头域
    putObjectRequest.addUserHeaders("x-obs-content-sha256", sha256Builder.toString());
    obsClient.putObject(putObjectRequest);
    System.out.println("PutObjectWithSha256 successfully");
} catch (ObsException e) {
    System.out.println("PutObjectWithSha256 failed");
    // 请求失败,打印http状态码
    System.out.println("HTTP Code:" + e.getResponseCode());
    // 请求失败,打印服务端错误码
    System.out.println("Error Code:" + e.getErrorCode());
    // 请求失败,打印详细错误信息
    System.out.println("Error Message:" + e.getErrorMessage());
    // 请求失败,打印请求id
    System.out.println("Request ID:" + e.getErrorRequestId());
    System.out.println("Host ID:" + e.getErrorHostId());
    e.printStackTrace();
} catch (Exception e) {
    System.out.println("PutObjectWithSha256 failed");
    // 其他异常信息打印
    e.printStackTrace();
}
}
```

## 须知

Java SDK 同时支持MD5与SHA256校验，在安全上更推荐使用SHA256算法。



## 23.15 为什么 SDK 源码中包含 acs.amazonaws.com 关键字? (Java 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[Java SDK](#):

```
2 usages
172     public static final String ALL_USERS_URI = "http://acs.amazonaws.com/groups/global/AllUsers";
173
2 usages
174     public static final String AUTHENTICATED_USERS_URI = "http://acs.amazonaws.com/groups/global/AuthenticatedUsers";
175
3 usages
176     public static final String LOG_DELIVERY_URI = "http://acs.amazonaws.com/groups/s3/LogDelivery";
```

## 23.16 如何理解 Content-Type ( MIME ) ? (Java 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 23-1 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

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.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
.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

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document	.dot	application/msword
.dotx	application/vnd.openxmlformats-officedocument.wordprocessingml.template	.drw	application/x-drw
.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

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.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
.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/javascript
.jsp	text/html	.la1	audio/x-liquid-file
.lar	application/x-laplayer-reg	.latex	application/latex
.lavs	audio/x-liquid-secure	.lbm	application/x-lbm
.lmsff	audio/x-la-lms	.ls	application/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

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.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
.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
.mxp	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

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.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
.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

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.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
.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-realtex
.rtf	application/msword	.rtf	application/x-rtf
.rv	video/vnd.rn-realvideo	.sam	application/x-sam
.sat	application/x-sat	.sdp	application/sdp

文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.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
.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



文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.vml	text/xml	.vpg	application/x- vpeg005
.vsd	application/ vnd.visio	.vsd	application/x-vsd
.vss	application/ vnd.visio	.vst	application/ vnd.visio
.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

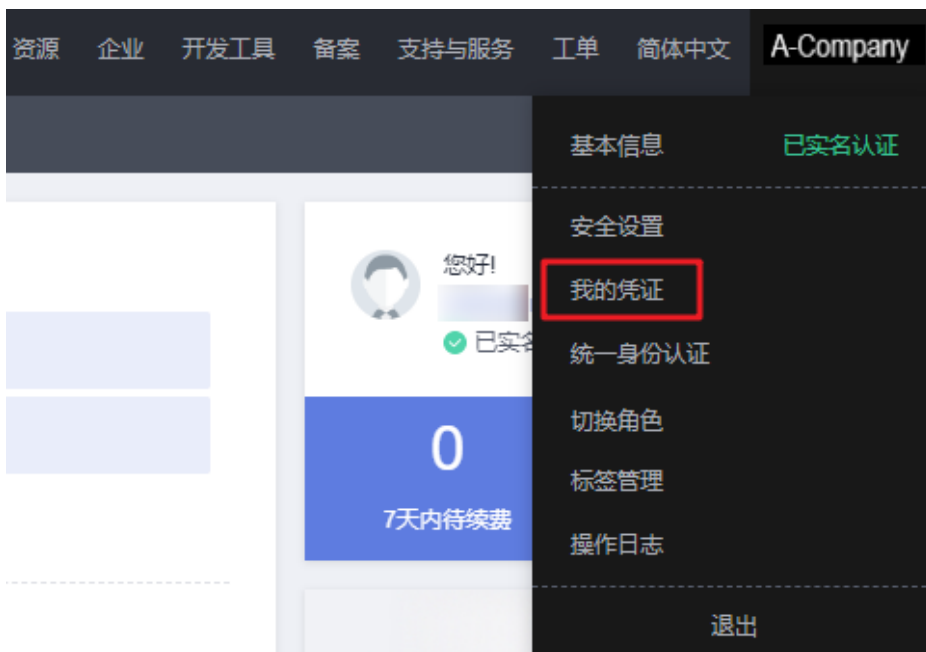
文件扩展名	Content-Type(Mime-Type)	文件扩展名	Content-Type(Mime-Type)
.x_b	application/x-x_b	.xdp	application/vnd.adobe.xdp
.xdr	text/xml	.xfd	application/vnd.adobe.xfd
.xdf	application/vnd.adobe.xfdf	.xhtml	text/html
.xls	application/vnd.ms-excel	.xls	application/x-xls
.xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	.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

## 23.17 如何获取账号 ID 和用户 ID?

### 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”。

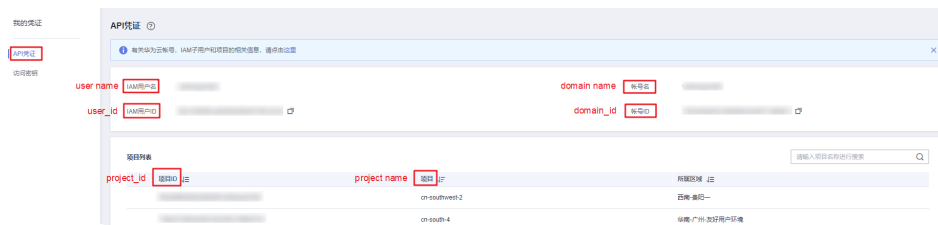
图 23-1 进入我的凭证



- 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 23-2 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。

图 23-3 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。

图 23-4 查看区域 ID



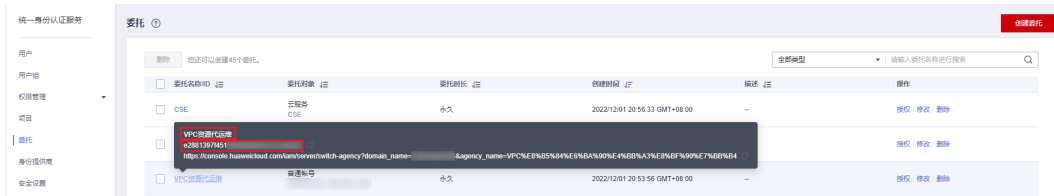
----结束

## 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。

图 23-5 查看委托 ID



----结束

# A 修订记录(Java SDK)

发布日期	修订记录
2024-04-18	第三十三次正式发布： <ul style="list-style-type: none"><li>• 新增章节<a href="#">设置对象AccessLabel</a>。</li><li>• 新增章节<a href="#">获取对象AccessLabel</a>。</li><li>• 新增章节<a href="#">删除对象AccessLabel</a>。</li></ul>
2024-01-12	第三十二次正式发布： <ul style="list-style-type: none"><li>• <a href="#">设置对象元数据(Java SDK)</a>章节增加接口约束。</li></ul>
2023-12-26	第三十一次正式发布： <ul style="list-style-type: none"><li>• 优化<a href="#">使用前须知(Java SDK)</a>、<a href="#">Java SDK接口概览(Java SDK)</a>、<a href="#">使用前准备(Java SDK)</a>、<a href="#">下载与安装SDK(Java SDK)</a>、<a href="#">快速入门(Java SDK)</a>、<a href="#">初始化(Java SDK)</a>章节内容。</li><li>• 优化<a href="#">管理对象(Java SDK)</a>、<a href="#">临时授权访问(Java SDK)</a>、<a href="#">多版本控制(Java SDK)</a>、<a href="#">生命周期管理(Java SDK)</a>、<a href="#">跨域资源共享(Java SDK)</a>、<a href="#">设置访问日志(Java SDK)</a>、<a href="#">静态网站托管(Java SDK)</a>、<a href="#">桶标签管理(Java SDK)</a>章节接口功能描述、接口约束及示例代码。</li></ul>
2023-12-15	第三十次正式发布： <ul style="list-style-type: none"><li>• 优化<a href="#">上传对象(Java SDK)</a>章节接口功能描述、接口约束及示例代码。</li><li>• 优化<a href="#">下载对象(Java SDK)</a>章节接口功能描述、接口约束及示例代码。</li></ul>
2023-11-25	第二十九次正式发布： <ul style="list-style-type: none"><li>• 优化<a href="#">管理并行文件系统(Java SDK)</a>章节接口功能描述、接口约束及示例代码。</li></ul>
2023-11-20	第二十八次正式发布： <ul style="list-style-type: none"><li>• 优化<a href="#">桶清单(Java SDK)</a>章节接口功能描述、接口约束及示例代码。</li></ul>

发布日期	修订记录
2023-11-14	第二十七次正式发布： <ul style="list-style-type: none"> <li>优化<b>管理桶(Java SDK)</b>章节接口功能描述、接口约束及示例代码。</li> </ul>
2023-08-26	第二十六次正式发布： <ul style="list-style-type: none"> <li>新增章节<b>如何指定Content-SHA256?(Java SDK)</b>。</li> <li>新增章节<b>为什么SDK源码中包含acs.amazonaws.com关键字? (Java SDK)</b>。</li> </ul>
2023-08-24	第二十五次正式发布。 新增章节： <ul style="list-style-type: none"> <li><b>通过Header鉴权访问OBS(Java SDK)</b></li> </ul> 修改章节： <ul style="list-style-type: none"> <li>开发环境准备 增加约束与限制内容。</li> </ul>
2023-08-10	第二十四次正式发布。 修改章节： <ul style="list-style-type: none"> <li><b>服务端加密(Java SDK)</b> 增加SSE-OBS加密类型说明。</li> </ul>
2023-08-07	第二十三次正式发布。 新增章节： <ul style="list-style-type: none"> <li>列举并行文件系统内对象</li> </ul>
2021-10-15	第二十二次正式发布 <ul style="list-style-type: none"> <li>修改安装 SDK 说明。</li> </ul>
2021-08-30	第二十一次正式发布。 <ul style="list-style-type: none"> <li>发布 3.21.8 版本。</li> </ul>
2021-07-30	第二十次正式发布。 新增章节： <ul style="list-style-type: none"> <li><b>依赖缺失和依赖冲突的解决</b></li> </ul>
2021-06-07	第十九次正式发布。 修改章节： <ul style="list-style-type: none"> <li>管理桶策略章节增加桶策略说明。</li> </ul>
2020-07-15	第十八次正式发布。 修改章节： <ul style="list-style-type: none"> <li><b>使用URL进行授权访问</b>，修复创建桶示例程序中的BUG。</li> </ul>

发布日期	修订记录
2020-06-30	第十七次正式发布。 修改章节： <ul style="list-style-type: none"><li>● 图片处理，新增通过临时授权方式实现图片处理的示例。</li><li>● 创建桶，新增创建多AZ桶的说明。</li></ul> 新增章节： <ul style="list-style-type: none"><li>● <a href="#">设置对象元数据</a></li></ul>
2020-02-11	第十六次正式发布。 新增章节： <ul style="list-style-type: none"><li>● <a href="#">如何生成SSE-C方式的加密密钥</a></li><li>● <a href="#">SDK是否支持批量上传、下载或者复制对象</a></li></ul>
2019-12-09	第十五次正式发布。 修改章节： <ul style="list-style-type: none"><li>● 创建OBS客户端，增加链式方式创建OBS客户端的相关说明。</li></ul>
2019-11-20	第十四次正式发布。 调整以下章节的位置： <ul style="list-style-type: none"><li>● 示例程序</li><li>● 技术支持渠道</li></ul>
2019-11-11	第十三次正式发布。 新增章节： <ul style="list-style-type: none"><li>● <a href="#">管理并行文件系统</a></li></ul>
2019-07-31	第十二次正式发布。 新增章节： <ul style="list-style-type: none"><li>● <a href="#">问题定位(Java SDK)</a></li><li>● <a href="#">HTTP状态码(Java SDK)</a></li><li>● <a href="#">常见问题(Java SDK)</a></li></ul>
2019-06-28	第十一次正式发布。 修改章节： <ul style="list-style-type: none"><li>● 示例程序，增加临时访问密钥相关示例代码。</li><li>● 安装SDK，增加gradle配置方式。</li></ul>

发布日期	修订记录
2019-03-30	第十次正式发布。 修改章节： <ul style="list-style-type: none"><li>● <a href="#">如何使对象可以被匿名用户访问?</a>，优化描述内容。</li><li>● 安装SDK，更新操作步骤。</li><li>● 10.2-OBS服务环境搭建，增加临时访问密钥相关内容。</li></ul> 新增章节： <ul style="list-style-type: none"><li>● API参考。</li></ul>
2019-03-05	第九次正式发布。 修改章节： <ul style="list-style-type: none"><li>● “事件通知”各子章节，新增函数 workflow 服务功能的说明及代码样例。</li></ul> 新增章节： <ul style="list-style-type: none"><li>● <a href="#">SDK的重试机制是什么?</a></li><li>● <a href="#">如何获取桶的静态网站访问地址?</a></li><li>● <a href="#">如何获取对象URL?</a></li><li>● <a href="#">如何快速上传大文件到OBS?</a></li></ul>
2019-02-18	第八次正式发布。 修改章节： <ul style="list-style-type: none"><li>● 使用URL进行授权访问，优化描述内容，增加“下载加密类型的对象”代码样例。</li><li>● 安装SDK，修改方式二的步骤。</li></ul>
2018-10-31	第七次正式发布。 新增章节： <ul style="list-style-type: none"><li>● 获取上传进度</li><li>● 获取下载进度</li></ul> 修改章节： <ul style="list-style-type: none"><li>● 配置OBS客户端，新增部分配置参数。</li></ul>
2018-08-31	第六次正式发布。 新增章节： <ul style="list-style-type: none"><li>● <a href="#">设置对象生命周期</a></li><li>● 基于表单上传</li></ul>
2018-06-11	第五次正式发布。 新增章节： <ul style="list-style-type: none"><li>● 追加上传</li></ul> 修改章节： <ul style="list-style-type: none"><li>● <a href="#">日志分析</a>，新增开启日志方式。</li></ul>



发布日期	修订记录
2018-01-31	<p>第四次正式发布。</p> <p>调整章节：</p> <ul style="list-style-type: none"><li>“快速入门”，新增OBS服务环境搭建、开发环境准备、安装SDK、获取服务地址。</li><li>“相关资源”，新增示例程序，移除OBS服务端环境搭建。</li><li>“初始化”，移除服务地址。</li></ul> <p>修改章节：</p> <ul style="list-style-type: none"><li>“临时授权访问”章节，修改示例代码。</li><li>“设置生命周期规则”章节，修改示例代码。</li><li>“查看生命周期规则”章节，修改示例代码。</li></ul>
2017-12-31	<p>第三次正式发布。</p> <p>新增章节：</p> <ul style="list-style-type: none"><li>图片处理</li></ul> <p>修改章节：</p> <ul style="list-style-type: none"><li>安装方式，新增Maven安装方式。</li><li><a href="#">生命周期管理</a>，新增对象转换策略特性。</li></ul>
2017-11-30	<p>第二次正式发布。</p> <p>新增章节：</p> <ul style="list-style-type: none"><li>桶存储类别</li><li>断点续传上传</li><li>断点续传下载</li></ul> <p>修改章节：</p> <ul style="list-style-type: none"><li><a href="#">设置对象属性</a>，新增设置对象存储类别相关内容</li></ul>
2017-10-10	<p>第一次正式发布。</p>