

对象存储服务

# API 参考

文档版本 15

发布日期 2018-04-25



#### 版权所有 © 华为技术有限公司 2018。 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默示的声明或保证。

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

### 华为技术有限公司

地址: 深圳市龙岗区坂田华为总部办公楼 邮编: 518129

网址: <a href="http://www.huawei.com">http://www.huawei.com</a>
客户服务邮箱: <a href="mailto:support@huawei.com">support@huawei.com</a>

客户服务电话: 4008302118

# 目录

1 文档简介	1
2 功能描述	2
3 公共消息头	6
4 错误响应消息	9
4.1 错误码列表	9
4.2 错误响应消息格式	14
5 请求鉴权方法	16
5.1 用户签名验证	
5.2 V2 普通请求鉴权	
5.3 V2 临时授权请求鉴权	
5.4 联邦认证鉴权	
6 权限控制	22
6.1 权限使用	
6.2 ACL	22
6.3 桶策略	25
6.4 角色	34
6.5 同时使用 ACL 和桶策略	34
6.6 FLASH 跨域访问配置	34
7 桶的基础操作	36
7.1 获取桶列表	36
7.2 创建桶	38
7.3 列举桶内对象	40
7.4 获取桶元数据	46
7.5 获取桶区域位置	47
7.6 获取桶存量信息	49
7.7 删除桶	50
8 桶策略配置	52
8.1 设置桶策略	53
8.2 设置桶配额	53
8.3 设置桶 ACL	

8.4 设置桶日志管理配置	57
8.5 设置桶的生命周期配置	59
8.6 设置桶的多版本状态	64
8.7 设置桶的消息通知配置	65
8.8 设置桶默认存储类型	70
8.9 设置桶的跨区域复制配置	71
8.10 获取桶策略	74
8.11 获取桶配额	75
8.12 获取桶 ACL	76
8.13 获取桶日志管理配置	78
8.14 获取桶的生命周期配置	80
8.15 获取桶的多版本状态	83
8.16 获取桶的消息通知配置	85
8.17 获取桶默认存储类型	88
8.18 获取桶的跨区域复制配置	89
8.19 删除桶策略	91
8.20 删除桶的生命周期配置	92
8.21 删除桶的跨区域复制配置	93
9 静态网站托管	95
9.1 设置桶的网站配置	95
9.2 设置桶的 CORS 配置	100
9.3 OPTIONS 桶	
9.4 OPTIONS 对象	
9.5 获取桶的网站配置	
9.6 获取桶的 CORS 配置	108
9.7 删除桶的网站配置	
9.8 删除桶的 CORS 配置	111
10 对象操作	113
10.1 PUT 上传	113
10.2 POST 上传	117
10.3 复制对象	
10.4 设置对象 ACL	
10.5 获取对象内容	
10.6 获取对象元数据	
10.7 获取对象 ACL	
10.8 删除对象	144
10.9 批量删除对象	146
10.10 取回归档存储对象	149
11 多段上传	152
11.1 初始化上传段任务	
11.2 列举多段上传任务	

11.3 上传段	161
11.4 拷贝段	
11.5 列举已上传的段	171
11.6 合并段	
11.7 取消多段上传任务	
12 服务端加密	180
12.1 服务端加密 SSE-KMS 方式	
12.2 服务端加密 SSE-C 方式	181
12.3 与服务端加密相关的接口	
13 并发一致性说明	185
A 文档修订记录	188

# **1** 文档简介

OBS通过简单的Web Services接口提供对象存储服务。通过对象存储服务可以构建基于存储的互联网应用,例如Web Hosting、在线视频、在线相册、在线备份等应用。OBS提供REST形式的访问接口,使用户能够非常容易地从Web应用中访问对象存储服务。用户可以通过本文档提供的简单的REST接口,在任何时间、任何地点、任何互联网设备上进行上传和下载数据。

在确保已充分了解OBS产品信息、使用协议以及收费方式后,再使用本文档所提到的接口。

# **2** 功能描述

OBS提供了一个基于Web服务接口,可随时在不同平台上使用web方式存储和检索任何数量的数据。OBS服务让所有开发人员都能访问同一个具备高扩展性、可靠性、安全性和快速价廉的数据存储基础设施。

OBS提供的交互接口遵循REST规范(HTTP 1.1版本)。使用OBS提供的REST接口,用户可以发送标准的HTTP请求完成创建、获取、删除桶或对象的操作。

#### OBS概念

在使用OBS前,必须了解主要的概念和术语,这些概念和术语如下:

#### 桶

桶用于存放对象的容器,是OBS中最高等级的命名空间。每个对象都存放在一个桶中。例如,如果名为"picture.jpg"的对象存放在"photo"桶中,则可使用 URL: http://photo.obs.cn-north-1.myhwclouds.com/picture.jpg对该对象进行寻址。您可以根据业务需要和网络位置来选择特定区域创桶。有关创桶详细信息,请参阅7.2 创建桶。

#### 对象

对象在OBS中是最基本的实体。在一个桶中可以存放多个对象,OBS系统并不能区分对象的类型。在OBS系统中存储的对象是被序列化了的,因此它可能是一个文本文件或者一个视频文件。OBS支持的数据大小范围可以是0B到48.8TB(包含0B和48.8TB),PutObject接口上传对象时对象最大为5GB,超过5GB对象需要使用多段方式上传。

#### 区域

OBS中的桶归属于某个区域,这个区域是您在创桶时选定的,桶一旦创建,其归属区域信息不能改变。您可以根据地理位置、降低成本、满足法规要求等标准来选择桶的区域。

有关OBS区域和终端节点的更多信息,请参考地区和终端节点。

#### M 4H AB

本文档所有示例和描述中使用的域名,均以华北-北京一区域的域名为例。实际使用过程中,请以业务的实际情况为准。

对象存储服务操作接口如下表所示。

#### 表 2-1 桶基础操作接口

接口	说明
GetAllMyBucket	获取用户的桶列表
CreateBucket	创建桶
ListBucket	列举桶中所有对象
HeadBucket	获取桶的元数据,测试桶的读权限。
GetBucketLocation	获取的桶的位置信息
GetBucketStorageInfo	获取桶存量信息
DeleteBucket	删除桶

#### 表 2-2 桶策略配置操作接口

接口	说明
PutBucketPolicy	设置桶的policy
PutBucketQuota	设置桶配额信息
PutBucketAcl	设置桶的ACL
PutBucketLogging	设置桶logging
PutLifecycleConfiguration	设置桶lifecycle配置
PutBucketVersioning	设置桶的多版本
PutBucketNotification	设置桶的通知配置
GetBucketPolicy	获取桶的policy信息
GetBucketQuota	获取桶配额信息
GetBucketAcl	获取桶的ACL
GetBucketLogging	获取桶logging的状态
GetLifecycleConfiguration	获取桶lifecycle配置
GetBucketVersioning	获取桶多版本状态
GetBucketNotification	获取桶的通知配置
DeleteBucketPolicy	删除桶的policy
DeleteLifecycleConfiguration	删除桶lifecycle配置

#### 表 2-3 静态网站托管操作接口

接口	说明
PutBucketWebsite	设置桶website
PutBucketCORS	设置桶的cors信息
OPTIONSBucket	OPTIONS桶
OPTIONSObject	OPTIONS对象
GetBucketWebsite	获取桶website信息
GetBucketCORS	获取桶的cors信息
DeleteBucketWebsite	删除桶website
DeleteBucketCORS	删除桶的cors信息

#### 表 2-4 对象操作接口

接口	说明
PutObject	PUT上传对象
PostObject	POST上传对象
CopyObject	复制对象
PutObjectAcl	设置对象ACL
GetObject	获取对象内容
HeadObject	获取对象元数据
GetObjectAcl	获取对象ACL
DeleteObject	删除对象
DeleteMultipleObject	批量删除对象
PostRestoreObject	取回归档对象

#### 表 2-5 多段上传操作接口

接口	说明
InitiateMultipartUpload	初始化上传段对象
ListBucketMultipartUploads	列举桶中的多段任务
UploadPart	上传段
UploadPart-Copy	拷贝段

接口	说明
ListParts	列举已上传的段
CompleteMultipartUpload	合并段
AbortMultipartUpload	取消多段上传任务

# **3** 公共消息头

REST公共消息头包含两类:公共请求消息头和公共响应消息头,本章对这部分内容进行了介绍。

#### 公共请求消息头

OBS系统对象服务的公共请求消息头如表3-1所示。

#### 表 3-1 公共请求消息头

消息头名称	描述	是否必选
Authorization	请求消息中可带的鉴权信息。 类型:字符串。 默认值:无。 条件:匿名请求不需要带,其他请求必选。	有条件必选
Content-Length	RFC 2616中定义的消息(不包含消息头)长度。 类型:字符串。 默认值:无。 条件:PUT操作和加载XML的操作必须带。	有条件必选
Content-Type	资源内容的类型,例如: text/plain。 类型:字符串。 默认值:无。	否
Date	请求发起端的日期和时间。 类型:字符串。 默认值:无。 条件:如果是匿名请求或者消息头中带了x- amz-date字段,则可以不带该字段,其他情况下必选。	有条件必选

消息头名称	描述	是否必选
Host	表明主机地址。如bucketname.obs.cn-north-1.myhwclouds.com。 类型:字符串。 默认值:无。	是
x-amz-security-token	用于识别联邦认证用户请求的头域。在联邦 认证功能开启的前提下,若携带此头域,则 将用户识别为联邦用户。 类型:字符串	否。当联邦 用户请求时 必须携带。

#### 公共响应消息头

OBS系统对象服务的公共响应消息头如表3-2所示。

#### 表 3-2 公共响应消息头

消息头名称	描述
Content-Length	响应消息体的字节长度。 类型:字符串。 默认值:无。
Connection	指明与服务器的连接是打开的还是关闭的。 类型:字符串。 有效值:keep-alive   close。 默认值:无。
Date	OBS系统响应的时间。 类型:字符串。 默认值:无。
ETag	对象的Hash值,只与对象内容有关,与对象元数据无关。 类型:字符串。
x-amz-id-2	帮助定位问题的特殊符号。 类型:字符串。 默认值:无。
x-amz-request-id	由OBS创建来唯一确定本次请求的值,可以通过该值来定位问题。 类型:字符串。 默认值:无。

消息头名称	描述
x-reserved	用来对版权进行说明。 类型:字符串。
	默认值:无。

# 4 错误响应消息

本章描述了OBS系统返回的对象存储访问错误响应消息,包括所有的错误码列表和错误响应消息的格式。

- 4.1 错误码列表
- 4.2 错误响应消息格式

# 4.1 错误码列表

在向OBS系统发出请求后,如果遇到错误,会在响应中包含响应的错误码描述错误信息。对象存储访问服务的错误码如表4-1所示。

#### 表 4-1 对象存储访问服务错误码列表

错误码	描述	HTTP状态码
AccessDenied	拒绝访问。	403 Forbidden
AccessForbidden	权限不足。	403 Forbidden
AccountProblem	用户的帐户出现异常(过期、冻结等),不能成功地完成操作。	403 Forbidden
AllAccessDisabled	用户无权限执行某操作。	403 Forbidden
AmbiguousGrantByEmailAddress	用户提供的Email地址关联的帐户超过了1个。	400 Bad Request
BadDigest	客户端指定的对象内容的 MD5值与系统接收到的内容 MD5值不一致。	400 Bad Request
BadDomainName	域名不合法。	400 Bad Request
BadRequest	请求参数不合法。	400 Bad Request

错误码	描述	HTTP状态码
BucketAlreadyExists	请求的桶名已经存在。桶的 命名空间是系统中所有用户 共用的,选择一个不同的桶 名再重试一次。	409 Conflict
BucketAlreadyOwnedByYou	发起该请求的用户已经创建 过了这个名字的桶,并拥有 这个桶。	409 Conflict
BucketNotEmpty	用户尝试删除的桶不为空。	409 Conflict
CredentialsNotSupported	该请求不支持证书验证。	400 Bad Request
CustomDomainAreadyExist	配置了已存在的域。	400 Bad Request
CustomDomainNotExist	操作的域不存在。	400 Bad Request
DeregisterUserId	用户已经注销。	403 Forbidden
EntityTooSmall	用户试图上传的对象大小小 于系统允许的最小大小。	400 Bad Request
EntityTooLarge	用户试图上传的对象大小超 过了系统允许的最大大小。	400 Bad Request
FrozenUserId	用户被冻结。	403 Forbidden
IllegalVersioningConfiguration Exception	请求中的版本配置无效。	400 Bad Request
IllegalLocationConstraintException	配置了与所在Region不匹配 的区域限制。	400 Bad Request
InArrearOrInsufficientBalance	用户欠费或余额不足而没有 权限进行某种操作。	403 Forbidden
IncompleteBody	请求体不完整。	400 Bad Request
IncorrectNumberOfFilesInPost Request	每个POST请求都需要带一个 上传的文件。	400 Bad Request
InlineDataTooLarge	Inline Data超过了允许的最大长度。	400 Bad Request
InsufficientStorageSpace	存储空间不足。	403 Forbidden
InternalError	系统遇到内部错误,请重 试。	500 Internal Server Error
InvalidAccessKeyId	系统记录中不存在客户提供 的Access Key Id。	403 Forbidden
InvalidAddressingHeader	用户必须指定匿名角色。	N/A
InvalidArgument	无效的参数。	400 Bad Request
InvalidBucketName	请求中指定的桶名无效。	400 Bad Request

错误码	描述	HTTP状态码
InvalidBucket	请求访问的桶已不存在。	400 Bad Request
InvalidBucketState	无效的桶状态。	409 Conflict
InvalidBucketStoragePolicy	修改桶策略时,提供的新策 略不合法。	400 Bad Request
InvalidDigest	HTTP头中指定的Content-MD5值无效。	400 Bad Request
InvalidEncryptionAlgorithmError	错误的加密算法。	400 Bad Request
InvalidLocationConstraint	创建桶时,指定的location不 合法。	400 Bad Request
InvalidPart	一个或多个指定的段无法找到。这些段可能没有上传,或者指定的entity tag与段的entity tag不一致。	400 Bad Request
InvalidPartOrder	段列表的顺序不是升序,段 列表必须按段号升序排列。	400 Bad Request
InvalidPayer	所有对这个对象的访问已经 无效了。	403 Forbidden
InvalidPolicyDocument	表单中的内容与策略文档中 指定的条件不一致。	400 Bad Request
InvalidRange	请求的range不可获得。	416 Client Requested Range Not Satisfiable
InvalidRedirectLocation	无效的重定向地址。	400 Bad Request
InvalidRequest	无效请求。	400 Bad Request
InvalidRequestBody	POST请求体无效。	400 Bad Request
InvalidSecurity	提供的安全证书无效。	403 Forbidden
InvalidStorageClass	用户指定的Storage Class无效。	400 Bad Request
InvalidTargetBucketForLogging	delivery group对目标桶无ACL权限。	400 Bad Request
InvalidURI	无法解析指定的URI。	400 Bad Request
KeyTooLong	提供的Key过长。	400 Bad Request
MalformedACLError	提供的XML格式错误,或者 不符合我们要求的格式。	400 Bad Request
MalformedError	请求中携带的XML格式不正 确。	400 Bad Request

错误码	描述	HTTP状态码
MalformedLoggingStatus	Logging的XML格式不正确。	400 Bad Request
MalformedPolicy	Bucket policy检查不通过。	400 Bad Request
MalformedPOSTRequest	POST请求的请求体不是结构 化良好的多段或形式化数 据。	400 Bad Request
MalformedQuotaError	Quota的XML格式不正确。	400 Bad Request
MalformedXML	当用户发送了一个配置项的 错误格式的XML会出现这样 的错误。	400 Bad Request
MaxMessageLengthExceeded	请求消息过长。	400 Bad Request
MaxPostPreDataLengthExceeded Error	在上传文件前面的POST请求 域过大。	400 Bad Request
MetadataTooLarge	元数据消息头超过了允许的 最大元数据大小。	400 Bad Request
MethodNotAllowed	指定的方法不允许操作在请 求的资源上。	405 Method Not Allowed
MissingContentLength	必须要提供HTTP消息头中的 Content-Length字段。	411 Length Required
MissingRegion	请求中缺少Region信息,且 系统无默认Region。	400 Bad Request
MissingRequestBodyError	当用户发送一个空的XML文 档作为请求时会发生。	400 Bad Request
MissingRequiredHeader	请求中缺少必要的头域。	400 Bad Request
MissingSecurityHeader	请求缺少一个必须的头。	400 Bad Request
NoSuchBucket	指定的桶不存在。	404 Not Found
NoSuchBucketPolicy	桶policy不存在。	404 Not Found
NoSuchCORSConfiguration	CORS配置不存在。	404 Not Found
NoSuchCustomDomain	请求的用户域不存在。	404 Not Found
NoSuchKey	指定的Key不存在。	404 Not Found
NoSuchLifecycleConfiguration	请求的LifeCycle不存在。	404 Not Found
NoSuchPolicy	给定的policy名字不存在。	404 Not Found
NoSuchUpload	指定的多段上传不存在。 Upload ID不存在,或者多段 上传已经终止或完成。	404 Not Found

错误码	描述	HTTP状态码
NoSuchVersion	请求中指定的version ID与现存的所有版本都不匹配。	404 Not Found
NoSuchWebsiteConfiguration	请求的Website不存在。	404 Not Found
NotImplemented	用户提供的消息头功能上还 没有实现。	501 Not Implemented
NotSignedUp	你的帐户还没有在系统中注 册,必须先在系统中注册了 才能使用该帐户。	403 Forbidden
OperationAborted	另外一个冲突的操作当前正 作用在这个资源上,请重 试。	409 Conflict
PermanentRedirect	尝试访问的桶必须使用指定 的结点,请将以后的请求发 送到这个结点。	301 Moved Permanently
PreconditionFailed	用户指定的先决条件中至少 有一项没有包含。	412 Precondition Failed
Redirect	临时重定向。	307 Moved Temporarily
RequestIsNotMultiPartContent	桶POST必须是闭式的多段/ 表单数据。	400 Bad Request
RequestTimeout	用户与Server之间的socket连 接在超时时间内没有进行读 写操作。	400 Bad Request
RequestTimeTooSkewed	请求的时间与服务器的时间 相差太大。	403 Forbidden
RequestTorrentOfBucketError	不允许请求桶的torrent文 件。	400 Bad Request
ServiceNotImplemented	请求的方法服务端没有实 现。	501 Not Implemented
ServiceNotSupported	请求的方法服务端不支持。	409 Conflict
ServiceUnavailable	服务器过载或者内部错误异常。	503 Service Unavailable
SignatureDoesNotMatch	请求中带的签名与系统计算 得到的签名不一致。检查你 的Secret Access Key 和签名 计算方法。	403 Forbidden
SlowDown	请降低请求频率。	503 Service Unavailable
System Capacity Not enough	系统空间不足异常 。	403 Forbidden

错误码	描述	HTTP状态码
TooManyCustomDomains	配置了过多的用户域	400 Bad Request
TemporaryRedirect	当DNS更新时,请求将被重 定向到桶。	307 Moved Temporarily
TooManyBuckets	用户拥有的桶的数量达到了 系统的上限,并且请求试图 创建一个新桶。	400 Bad Request
TooManyObjectCopied	用户单个对象被拷贝的数量 超过系统上限。	400 Bad Request
TooManyWrongSignature	因高频错误请求被拒绝服 务。	400 Bad Request
UnexpectedContent	该请求不支持带内容字段。	400 Bad Request
UnresolvableGrantByEmailAddress	用户提供的Email与记录中任 何帐户的都不匹配。	400 Bad Request
UserKeyMustBeSpecified	请求中缺少用户的AK信息。	400 Bad Request
WebsiteRedirect	Website请求缺少 bucketName。	301 Moved Permanently
KMS.DisabledException	SSE-KMS加密方式下,主密 钥被禁用。	400 Bad Request
KMS.NotFoundException	SSE-KMS加密方式下,主密 钥不存在。	400 Bad Request

# 4.2 错误响应消息格式

#### 错误响应消息头

当错误发生时,消息头中都会包含:

- Content-Type: application/xml.
- 对应的 3xx, 4xx, 或 5xx HTTP 状态码, 具体的错误码请参见表4-1。

#### 错误响应消息体

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

<?xml version="1.0" encoding="UTF-8"?>

<Error>

<Code>NoSuchKey</Code>

<Message>The resource you requested does not exist/Message>

<Resource>/example-bucket/object</Resource>

<RequestId>001B21A61C6C0000013402C4616D5285/RequestId>

<HostId>RkRCRDJENDc5MzdGQkQ40UY3MTI4NTQ3NDk2Mjg0M0FB

QUFBQUFBYmJiYmJiYmJD</HostId>

</Error

#### 错误响应消息元素

REST错误响应消息元素如表4-2所示。

#### 表 4-2 错误响应消息元素

元素名称	描述	
Error	错误响应消息体XML结构中描述错误信息的根节点元素。	
Code	错误响应消息体XML中错误响应对应的HTTP消息返回码,具体的错误码请参见表4-1。	
Message	错误响应消息体XML中具体错误更全面、详细的英文解释,具体的错误消息请参见表4-1。	
RequestId	本次错误请求的请求ID,用于错误定位。	
HostId	返回该消息的服务端ID。	
Resource	该错误相关的桶或对象资源。	

#### ∭说明

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

# 5 请求鉴权方法

- 5.1 用户签名验证
- 5.2 V2普通请求鉴权
- 5.3 V2临时授权请求鉴权
- 5.4 联邦认证鉴权

## 5.1 用户签名验证

OBS向用户提供的帐户包括了AK(Access Key ID)和SK(Secret Access Key),用于鉴别用户身份。在向OBS系统发送请求时,客户端发送的每个消息头需要包含由SK、请求时间、请求类型等信息生成的鉴权信息。

OBS根据应用场景,提供了普通鉴权方式、基于表单格式上传对象鉴权方式和临时鉴权方式。

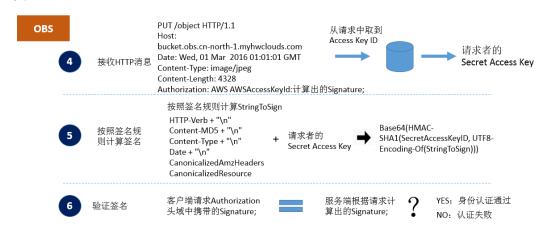
OBS签名计算流程如图5-1和图5-2所示:

#### 图 5-1 OBS 签名计算流程(1)



- 1. 客户端在调用OBS API接口前,根据OBS定义的API接口格式,构造HTTP消息。消息构造完成后,客户端可以根据签名计算规则,从HTTP消息中取特定的字符串,构造出签名所需的StringToSign。
- 2. 客户端使用自己的Secret Access Key和构造出的StringToSign计算出签名字符串。
- 3. 客户端在构造出HTTP消息头域中增加一个签名头域,将用户的Access Key和计算出的签名字符串放在签名头域中。客户端将携带签名头域的HTTP消息发送给OBS,等待OBS验证结果。

#### 图 5-2 OBS 签名计算流程(2)



- 4. OBS服务端收到客户端发来的请求后,先根据签名头域中的Access Key取回用户的Secret Access Key。
- 5. OBS服务端根据签名计算规则,从HTTP消息中取特定的字符串,构造出签名所需的StringToSign。OBS将Secret Access Key和服务端构造出的StringToSign计算得到签名字符串。
- 6. OBS用服务端计算出的签名字符串和请求头域中携带的客户端计算出的签名字符串做对比。如果服务端和客户端签名一致,则说明客户端使用的Secret Access Key正确,客户端的身份可以被确认,OBS会按照接口定义进行后续处理。如服务端和客户端签名不一致,说明客户端使用的Secret Access Key不正确,客户端的身份不能被确认,OBS会直接拒绝该请求。

### 5.2 V2 普通请求鉴权

OBS的所有API接口都可以通过普通请求鉴权方式来认证身份,也是最常用的身份认证方式。

普通请求鉴权是指将通过HTTP/HTTPS消息头Authorization进行鉴权, V2普通请求鉴权消息的格式为:

Authorization: AWS AccessKeyID:signature.

鉴权消息(signature)生成过程如下:

1. 构造Request字符串,参数说明如**表5-1**所示。
StringToSign = HTTP-Verb + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Date + "\n" + CanonicalizedOBSHeaders + CanonicalizedResource。

表 5-1 构造 Request 所需参数说明

参数	描述	
HTTP-Verb	指接口操作的方法,对REST接口而言,即为http请求操作的VERB,如: "PUT","GET","DELETE"等字符串。	
Date	生成请求的时间,该时间格式遵循RFC 1123; 当有自定义字段x-amz-date时,该参数按照空字符串处理;见表 5-3。 如果进行临时授权方式操作(如临时授权方式获取对象内容等操 作)时,该参数不需要。	
Content-Type	内容类型,用于指定消息类型,例如: text/plain。 当请求中不带该头域时,该参数按照空字符串处理,见 <b>表5-2</b> 。	
Content- MD5	按照RFC 1864标准计算出消息体的MD5摘要字符串,即消息体 128-bit MD5值经过base64编码后得到的字符串。	
Canonicalize dOBSHeader s	OBS自定义的字段,以"x-amz-"作为前辍的消息头,如"x-amz-date, x-amz-acl"。  1.自定义字段中的所有字符要转为小写,需要添加多个字段时,要将所有字段按照字典序进行排序。  2.在添加自定义字段时,如果有重名的字段,则需要进行合并。如: x-amz-meta-name:name1和x-amz-meta-name:name2,则需要合并成x-amze-meta-name:name1,name2。  3.当自定义字段中,含有非ASCII码或不可识别字符时,需进行Base64编码  4.当自定义字段中含有无意义空格或table键时,需要摒弃。例如: x-amz-meta-name: name(name前带有一个无意义空格),需要转换为: x-amz-meta-name:name  5.每一个自定义字段最后都需要另起新行,见表5-4	
Canonicalize dResource	表示HTTP请求所指定的OBS资源,构造方式如下: <桶名+对象名>+[子资源]+[查询字符串] 1.桶名和对象名,例如:/bucket/object.如果没有对象名,如列举桶。则为"/bucket/",如桶名也没有,则为"/" 2.如果有子资源,则将子资源添加进来,例如?acl,?logging。子资源包括acl, lifecycle, location, logging, notification, partNumber, policy, uploadId, uploads, versionId, versioning, versions, website, quota, storagePolicy, storageinfo, delete。 3.如有查询字符串那么将这些查询字符串及其请求值按照字典序从小到大排列。	

生成StringToSign的例子:

#### 表 5-2 获取对象

请求消息头	StringToSign
GET /object.txt HTTP/1.1	GET \n
Host: bucketname.obs.cn-	\n
north-1.myhwclouds.com	Sat, 12 Oct 2015 08:12:38 GMT\n
Date: Sat, 12 Oct 2015 08:12:38 GMT	/bucket/object.txt

#### 表 5-3 带自定义字段上传对象 1

请求消息头	StringToSign
PUT /object.txt HTTP/1.1	PUT\n
User-Agent: curl/7.15.5	\n
Host: bucketname.obs.cn- north-1.myhwclouds.com	text/plain\n x-amz-date:Tue, 15 Oct 2015 07:20:09
x-amz-date:Tue, 15 Oct 2015 07:20:09	GMT\n
GMT	/bucket/object.txt
content-type: text/plain	·
Content-Length: 5913339	

#### 表 5-4 带自定义字段上传对象 2

请求消息头	StringToSign
PUT /object.txt HTTP/1.1 User-Agent: curl/7.15.5	PUT\n \n
Host: bucketname.obs.cn- north-1.myhwclouds.com Date: Mon, 14 Oct 2015 12:08:34 GMT x-amz-acl: public-read content-type: text/plain Content-Length: 5913339	text/plain\n Mon, 14 Oct 2015 12:08:34 GMT\n x-amz-acl:public-read\n /bucket/object.txt

#### 表 5-5 获取对象 ACL

请求消息头	StringToSign
GET /object.txt?acl HTTP/1.1	GET \n
Host: bucketname.obs.cn-	\n
north-1.myhwclouds.com	Sat, 12 Oct 2015 08:12:38 GMT\n
Date: Sat, 12 Oct 2015 08:12:38 GMT	/bucket/object.txt?acl

2. 根据Request和SK生成signature, 生成过程使用HMAC(hash-based authentication code)算法,

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

## 5.3 V2 临时授权请求鉴权

临时授权(Query String签名)请求: OBS服务支持已经注册且激活的用户用自己的账户,构造一个特定操作的URL,这个URL中会包含鉴权信息,任何拿到这个URL的用户均可执行这个操作。例如临时授权方式下载对象,拿到相应URL的用户都能下载这个对象,但该URL只在Expires指定的失效时间内有效。临时授权请求鉴权方式,主要用于在不提供给其他人Secret Access Key的情况下,让其他人能用自己的身份执行自己定义的操作。

V2临时授权请求的格式如下:

GET /ObjectKey?AWSAccessKeyId=AccessKeyID&Expires=ExpiresValue&Signature=signature HTTP/1.1 Host: bucketname.obs.cn-north-1.myhwclouds.com

临时授权请求通过消息参数指定临时鉴权信息,参数具体意义如表5-6所示。

表 5-6	临时授权请求消息参数
1X J-U	111111111111111111111111111111111111111

参数名称	描述	是否必选
AWSAccessKeyId	授权者的AK信息。 类型:字符串。	是
Expires	临时授权失效的时间;UTC时间,1970年1月1日零时之后的指定的Expires时间内有效(以秒为单位)。 类型:字符串。	是
Signature	根据用户SK、Expires等参数计算出的签名信息。 类型:字符串。	是

临时授权方式的Signature计算方法和Authorization消息头有两处不同:

- 临时授权方式Base64编码后的签名信息还要经过URL编码。
- StringToSign中的Expires和原来Authorization消息中的消息头Date对应。

 $StringToSign = HTTP-Verb + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Expire + "\n" + CanonicalizedOBSHeaders + CanonicalizedResource.$ 

Signature = URL-Encode (Base64 (  $\,$  HMAC-SHA1 (  $\,$  UTF-8-Encoding-0f (YourSecretAccessKeyID, StringToSign ) ))).

### 5.4 联邦认证鉴权

OBS支持联邦认证用户访问OBS,访问时必须同时携带V2的签名信息和security token。签名信息中的ak和security token是由IAM为联邦认证用户颁发的。签名信息计算方法和V2普通鉴权的计算方法一样。

联邦认证用户发送请求时必须携带如下格式的身份认证信息:

Authorization: AWS AKIAIOSFODNN7EXAMPLE:QBaO+tS/76QYHVnUoxvf9EPH/3o=x-amz-security-token: security token string

如果一个OBS请求消息中携带了x-amz-security-token时,OBS默认该请求是联邦证用户发送请求,并按照联邦认证鉴权鉴权方法进行验证。如果签名或者security token有一个不正确,本次请求的身份认证都会失败。

# 6 权限控制

本章描述OBS系统使用ACL、Bucket Policy控制用户访问桶或对象资源的访问权限。

- 6.1 权限使用
- 6.2 ACL
- 6.3 桶策略
- 6.4 角色
- 6.5 同时使用ACL和桶策略
- 6.6 FLASH跨域访问配置

## 6.1 权限使用

权限控制用来管理请求者对请求资源(桶或者对象)的访问权限,请求用户只能访问 自己被授权的资源,如果访问未被授权的资源,则请求失败。

用户创建的桶或者对象,其所有者为这个用户所属的domain。默认情况下,只有这个domain下的用户对这个桶或者资源有访问权限。可以通过ACL(Access Control List)或者Bucket Policy两种方式管理权限

- ACL:使用权限控制列表授予某个账户对资源的访问权限,每个表项指定一个确定的权限给指定的账户;ACL只能授予账户(Account),不能授予Account下的用户(User);ACL只能用来授予权限,不能禁止权限;
- Bucket Policy: 控制单个用户或多个用户访问桶或桶下对象的访问权限; Bucket Policy既可以用来授予权限,也可以禁止权限; 桶策略即可作用于Account, 也可作用于User; 需要注意的是, 如果要设置一个Account创建的桶policy, 只能由这个Account本身和它账户下的User。当赋予某个User设置policy的权限时, OBS返回200; 当User没有权限设置policy, OBS返回405。

#### 6.2 ACL

每个桶或者对象在创建时都会生成默认的ACL, ACL列表的每项包含了对被授权用户授予什么样的权限。对存在的桶或对象也可以调用更改ACL接口生成新的ACL。

● 被授权用户

#### 表 6-1 OBS 支持的被授权用户

被授权用户	描述	
OBS用户	桶或对象的权限可以授予任何拥有OBS帐户的用户,被授权后对应的OBS用户可以使用AK和SK访问OBS。	
注册用户组	桶或对象的权限还可以授予拥有注册用户组帐户的用户,被授权后所有的注册用户组的用户可以使用AK和SK访问OBS。所有的OBS用户都属于注册用户组。	
匿名用户	如果匿名用户被授予了访问桶或对象的权限,所有人都可以访问对应的桶或对象。	
日志投递用 户组	桶的权限可以授予给日志投递的用户,被授权后该组类的用户可以访问该桶。该权限一般用在桶日志管理配置中。	

#### ● ACL格式

在执行PUT /?acl, PUT /ObjectKey?acl操作时,需要在请求消息体中包含ACL,ACL的格式如下所示:

```
<AccessControlPolicy>
<Owner>
<ID>id</ID>
<DisplayName>displayname</DisplayName>
</Owner>
<AccessControlList>
<Grant>
<Grantee>grantee</Grantee>
<Permission>permission</Permission>
</Grant>
<Grant>
</AccessControlList>
</AccessControlList>
</AccessControlList>
</AccessControlList>
</AccessControlList>
</AccessControlList>
</AccessControlPolicy>
```

其中permission是OBS支持的5种权限(详见表6-2),<Grantee>节点按照不同的被授予权限用户包括OBS用户、注册用户组用户、匿名用户以及日志投递用户组用户,有不同的格式。

#### ● 授予OBS用户

<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
<ID>domainId</ID>
<DisplayName>displayname</DisplayName>
</Grantee>

#### ● 授予注册用户组

<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
<URI>http://acs.amazonaws.com/groups/global/AuthenticatedUsers</URI>
</Grantee>

#### ● 授予匿名用户

<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
<URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
</Grantee>

#### ● 日志投递用户组

<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
<URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
</Grantee>

#### 表 6-2 OBS 支持的桶或对象权限

权限	描述
READ	若有桶的读权限,则可以获取该桶内对象列表和桶的元数据。 若有对象的读权限,则可以获取该对象内容和元数据。
WRITE	若有桶的写权限,则可以上传、覆盖和删除该桶内任何对象。 此权限在对象上不适用。
READ_ACP	若有读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。
WRITE_ACP	若有写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。
FULL_CONTROL	若有桶的完全控制权限意味着拥有READ、WRITE、READ_ACP和WRITE_ACP的权限。
	若有对象的完全控制权限意味着拥有READ、READ_ACP和WRITE_ACP的权限。READ_ONLY的用户不受此限制。

#### □说明

在一个请求中最多支持100个授权。

每一次对桶或对象的授权操作都将覆盖桶或对象已有的权限列表,而不会对其新增权限。

#### 权限控制策略

OBS支持在创建桶或上传对象时通过x-amz-acl头域设置桶或对象的权限控制策略,其设置的权限控制策略只能选择预定义的几种策略,如表6-3所示。

#### 表 6-3 OBS 预定义的权限控制策略

预定义的权限控制策略	描述
private	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。
public-read	桶或对象的所有者拥有完全控制的权限,其他所有用户包括匿名用户拥有读的权限。
public-read-write	桶或对象的所有者拥有完全控制的权限,其 他所有用户包括匿名用户拥有读和写的权 限。
authenticated-read	桶或对象的所有者拥有完全控制的权限,其 他OBS授权用户拥有读权限。

预定义的权限控制策略	描述
bucket-owner-read	对象的所有者拥有完全控制的权限,桶的所 有者拥有只读的权限。
bucket-owner-full-control	对象的所有者拥有完全控制的权限,桶的所 有者拥有完全控制的权限。
log-delivery-write	日志投递用户组拥有对桶的写权限以及读 ACP的权限。

#### **山湖明**

系统默认权限控制策略为private权限。

# 6.3 桶策略

桶策略可基于各种条件,如OBS操作、申请人、资源和请求的其他要素(如IP地址) 提供对桶和对象的集中访问控制。附加到某个桶上的权限适用于该桶内所有对象。

个人和公司均可使用桶策略。当公司注册OBS时,会创建一个账户。即在OBS上,该公司等同于该账户。该账户在财务上为该公司及其员工所创建的OBS资源负责。账户可基于各种条件授予权限给桶策略和员工。例如,账户可以按以下维度制定策略,授予某用户的写权限:

- 给予用户特定的桶的写权限
- ◆ 给予特定的网络中的用户写权限
- 给予用户的自定程序写权限

访问控制列表(access control lists,ACL)只能对单个对象进行权限的添加,而桶策略可对一个桶内的所有对象进行权限的添加和禁止。账户可通过同一请求对某桶内任意数量的对象进行权限设置。此外,账户还可以对亚马逊资源名称(Amazon resource names,ARNs)及其他值添加通配符(类似于正则表达式运算符),从而实现对一组对象的访问控制。

Bucket Policy允许桶的所有者执行Put Bucket Policy操作,给Bucket设置访问控制策略,如果Bucket已有Policy,则新的Policy会覆盖老的Policy。Bucket Owner也可以执行Get Bucket Policy 或 Delete Bucket Policy操作,获取或删除已设置的Bucket Policy。设置 Policy后,后续对此Bucket的访问请求都会受到Policy的限制,表现为拒绝或接受请求。对拒绝或接受的判断,取决于Policy的描述。如下所示的Policy允许账户 783fc6652cf246c096ea836694f71855(域ID)和 219d520ceac84c5a98b237431a2cf4c2(域ID)对桶mybucket下的所有对象执行GetObject权限:

### Policy 格式

一个Policy由JSON描述,格式定义为:

```
{
  "Version" : "version",
  "Id" : "id",
  "Statement" : [statement]
}
```

其中Id和Version为可选字段:

#### 表 6-4 Policy 结构

元素	描述	
Id	可选关键字,Policy Id,描述Policy的字符串	
Version	可选关键字,目前version字段支持"2008-10-17"	

Statement包含若干条statement, 一条statement的结构包括:

#### 表 6-5 statement 结构

元素	描述	是否必选
Sid	statement Id,可选关键字,描述statement的字符串	可选
Principal	可选关键字,被授权人,指定本条statement权限针对的Domain或User,支持通配符"*",表示所有用户。当对Domain进行授权时,Principal支持三种格式AWS:domainid、AWS:arn:aws:iam::domainid:root或CanonicalUser:domainid。当对User进行授权时,Principal格式为AWS:arn:aws:iam::domainid:user/userId或者AWS:arn:aws:iam::domainid:user/userName。被授权人也可以指定本条statement权限针对的用户为Agency用户或者Federated用户。当对Agency用户进行授权时,Principal格式为AWS:arn:aws:iam::domainid:agency/agencyName。当对Federated用户授权时,Principal格式为Federated:arn:aws:iam::domainid:identity-provider/providername或者Federated:arn:aws:iam::domainid:group/groupname。	可选,Principal 与NotPrincipal选 其一

元素	描述	是否必选
NotPrincipal	可选关键字,不被授权人。指定本条statement权限作用于除该关键字指定Domain或User之外的授权人。取值同Principal。	可选, NotPrincipal与 Principal选其一
Action	可选关键字,指定本条statement作用的操作,Action 字段为OBS支持的所有操作集合,以字符串形式表示,不区分大小写。支持通配符"*",表示该资源能进行的所有操作。例如: "Action": ["s3:List*", "s3:Get*"]。	可选,Action与 NotAction选其 一
NotAction	可选关键字,指定一组操作,statement匹配除该组操作之外的其他操作。 取值同Action。	可选, NotAction与 Action选其一
Effect	必选关键字,效果,指定本条statement的权限是允许还是拒绝,Effect的值必须为Allow或者Deny。	必选
Resource	可选关键字,指定statement起作用的一组资源,资源以ARN(Amazon Resource Name)格式定义,支持通配符"*",表示所有资源。	可选,Resource 与NotResource 选其一
NotResource	可选关键字,指定一组资源,statement匹配除该组 资源之外的其他资源。 取值同Resource。	可选, NotResource与 Resource选其一
Condition	可选关键字,本条statement生效的条件。	可选

#### □ 说明

在单条statement中,Action与NotAction必须二选一,Resource与NotResource必须二选一,Principal与NotPrincipal必须二选一。

#### OBS支持的桶Action有:

- s3:CreateBucket
- s3:DeleteBucket
- s3:ListBucket
- s3:ListBucketVersions
- s3:ListBucketMultipartUploads
- s3:GetBucketAcl
- s3:PutBucketAcl
- s3:GetBucketCORS
- s3:PutBucketCORS
- s3:GetBucketVersioning
- s3:PutBucketVersioning
- s3:GetBucketLocation
- s3:GetBucketLogging

- s3:PutBucketLogging
- s3:GetBucketWebsite
- s3:PutBucketWebsite
- s3:DeleteBucketWebsite
- s3:GetLifecycleConfiguration
- s3:PutLifecycleConfiguration
- s3:GetBucketNotification
- s3:PutBucketNotification
- s3:GetReplicationConfiguration

#### OBS支持的对象Action有:

- s3:GetObject(可作用于GET Object, HEAD Object)
- s3:GetObjectVersion
- s3:PutObject(可作用于PUT Object, POST Object, Initiate Multipart Upload, Upload Part, Complete Multipart Upload)
- s3:GetObjectAcl
- s3:GetObjectVersionAcl
- s3:PutObjectAcl
- s3:PutObjectVersionAcl
- s3:DeleteObject
- s3:DeleteObjectVersion
- s3:ListMultipartUploadParts
- s3:AbortMultipartUpload
- s3:RestoreObject
- s3:ReplicateObject
- s3:ReplicateDelete

#### OBS支持操作S3 Resource的ARN表示:

arn:aws:s3:::bucketname (桶操作)

arn:aws:s3:::bucketname/path/objectname (对象操作)

以下示例策略向账户 b4bf1b36d9ca43d984fbcb9491b6fce9 (域ID)下的用户ID为71f3901173514e6988115ea2c26d1999的user1用户授予 examplebucket的所有操作权限(包含桶操作与对象操作)。

```
]
}
]
}
```

#### 或者

#### ∭说明

如果在上传对象时没有指定path,则ARN中省去"/path"。

Statement中可选的Condition判断类型如下表所示:

#### 表 6-6 Condition

类型	关键字	说明
String	StringEquals	字符串匹配,简化为: streq
	StringNotEquals	字符串不匹配,简化为: strneq
	StringEqualsIgnoreCase	忽略大小写的字符串匹配,简化为: streqi
	StringNotEqualsIgnoreCase	忽略大小写的字符串不匹配,简化为: strneqi
	StringLike	宽松的区分大小写的匹配。这些值可以 在字符串中的任何地方包括一个多字符 匹配的通配符(*)和单字符匹配通配符 (?)。简化为: strl
	StringNotLike	非宽松区分大小写的匹配。这些值可以 在字符串中的任何地方包括一个多字符 匹配的通配符(*)和单字符匹配通配符 (?)。简化为: strnl
Numeric	NumericEquals	相等,简化为: numeq
	NumericNotEquals	不相等,简化为: numneq
	NumericLessThan	小于,简化为: numlt
	NumericLessThanEquals	小于等于,简化为: numlteq
	NumericGreaterThan	大于,简化为: numgt
	NumericGreaterThanEquals	大于等于,简化为: numgteq

类型	关键字	说明
Date	DateEquals	日期时间相等,简化为: dateeq
	DateNotEquals	日期时间不相等,简化为: dateneq
	DateLessThan	日期时间小于,简化为: datelt
	DateLessThanEquals	日期时间小于等于,简化为: dateIteq
	DateGreaterThan	日期时间大于,简化为: dategt
	DateGreaterThanEquals	日期时间大于等于,简化为: dategteq
Boolean	Bool	严格布尔值相等
IP address	IpAddress	指定的IP或IP范围
	NotIpAddress	除指定的IP或IP范围外所有IP

#### ∭说明

Condition关键字区分大小写。Date格式符合ISO 8601 规范,参考http://www.w3.org/TR/NOTE-datetime

每个Condition块可以包含多个key-value的组合,如下示例中Condition组合表示的判断条件为:请求时间从2009-04-16T12:00:00Z到2009-04-16T15:00:00Z,请求的IP地址范围是192.168.176.0/24"或"192.168.143.0/24"网段的请求。

```
"Condition" : {
    "DateGreaterThan" : {
    "aws:CurrentTime" : "2009-04-16T12:00:00Z"
    },
    "DateLessThan": {
    "aws:CurrentTime" : "2009-04-16T15:00:00Z"
    },
    "IpAddress" : {
        "aws:SourceIp" : ["192.168.176.0/24", "192.168.143.0/24"]
    }
}
```

Condition块中可选的key包括两种:和Action无关的通用key、和Action有关系的key。和Action无关的key包括:

#### 表 6-7 Common Condition Key

Condition Key	Condition Type
aws:CurrentTime	Date
aws:EpochTime	Numeric
aws:SecureTransport	Bool
aws:SourceIp	IP address
aws:UserAgent	String

Condition Key	Condition Type
aws:Referer	String

和OBS Action有关的Key包括:

#### 表 6-8 OBS Action Condition Key

Action	可选Key	描述
s3:CreateBucket	s3:x-amz-acl	创建桶时在头域x-amz-acl 中可以包含的 Canned ACL 取值范围为private  public-read  public-read-write authenticated-read bucket- owner-read
		bucket-owner-full-control log-delivery-write.
	s3:x-amz-grant- permission	暂不支持
	s3:LocationConstrai nt	暂不支持
s3:ListBucket	s3:prefix	String类型
	s3:delimiter	String类型
	s3:max-keys	Numeric类型
s3:ListBucketVer	s3:prefix	String类型
sions	s3:delimiter	String类型
	s3:max-keys	Numeric类型
s3:PutBucketAcl	s3:x-amz-acl	修改桶ACL时在头域x-amz-acl 中可以包含的 Canned ACL 取值范围为private  public-read  public-read-write authenticated-read bucket- owner-read  bucket-owner-full-control log-delivery-write.
	s3:x-amz-grant- permission	暂不支持
s3:PutObject	s3:x-amz-acl	上传对象时在头域x-amz-acl 中可以包含的 Canned ACL 取值范围为private  public-read  public-read-write authenticated-read bucket- owner-read
		bucket-owner-full-control log-delivery-write.
	s3:x-amz-copy- source	String,格式如/bucketname/keyname
	s3:x-amz-metadata- directive	取值范围为 COPY REPLACE

Action	可选Key	描述
	s3:x-amz-grant- permission	暂不支持
	s3:x-amz-storage- class	暂不支持
s3:PutObjectAcl	s3:x-amz-acl	修改对象ACL时在头域x-amz-acl 中可以包含的Canned ACL 取值范围为private  public-read  public-read-write authenticated-read  bucket-owner-read
		bucket-owner-full-control log-delivery-write.
	s3:x-amz-grant- permission	暂不支持
s3:GetObjectVer sion	s3:VersionId	String类型
s3:GetObjectVer sionAcl	s3:VersionId	String类型
s3:PutObjectVers	s3:VersionId	String类型
ionAcl	s3:x-amz-acl	修改对象版本ACL时在头域x-amz-acl 中可以 包含的Canned ACL 取值范围为private  public-read  public-read-write authenticated- read bucket-owner-read
		bucket-owner-full-control log-delivery-write.
	s3:x-amz-grant- permission	暂不支持
s3:DeleteObject Version	s3:VersionId	String类型
s3:* (Actions or any of the S3 Actions)	s3:signatureversion	暂不支持
	s3:authType	暂不支持
	s3:signatureAge	暂不支持
	s3:x-amz-content- sha256	暂不支持

# Policy 权限判断逻辑

Policy在做权限判断时,每条statement会有3种结果,explicit deny、allow和default deny。Bucket Policy对于Policy中的多条statement采用以下规则进行判定:Bucket Policy对Policy中包含的每条statement都要进行Explicit Deny、Allow和Default Deny的判断,最终的判决结果遵循explicit deny>allow>default deny的规则;

- 1.如果没有显式的Deny和Allow,则请求权限判别为Default Deny
- 2.显式的Deny覆盖Allow;

- 3.Allow覆盖默认的Default Deny;
- 4.statement的顺序没有影响

#### 表 6-9 Statement Result

名称	说明
explicit deny	显式拒绝访问,资源匹配的statement中effect="deny",表明 Request无法进行访问,此时直接返回无权限失败。
allow	允许访问,资源匹配的statement中effect="allow",表明Request可以进行访问,继续下一条statement判断。
default deny	默认拒绝访问,在没有任何一条statement与Request匹配上,默认本次Request无法进行访问。

## 防盗链设置

OBS是按使用收费的服务,为了防止用户在OBS的数据被其他人盗链,OBS支持基于 HTTP header中表头字段referer的防盗链方法。OBS同时支持访问白名单和访问黑名单 的设置。

#### ● 白名单设置

通过设置白名单,用户可以指定允许来自白名单列表中的网站的请求访问,否则将进行拦截。

通过浏览器地址栏直接访问资源的请求,即HTTP请求中referer为空的场景,用户可以通过在Condition的"aws:Referer"中添加\${null}字段,用于指定是否允许referer为空的请求访问。

白名单设置具体参考如下policy设置进行实现:

如果按照此方式设置,只有referer为"www.example01.com"和referer为空的请求可以对桶bucket中的资源进行操作。

#### ● 黑名单设置

使用访问黑名单功能可以参考如下policy设置进行实现:

```
"Statement":[
    {"Sid":"1",
    "Effect":"Deny",
    "Principal":{"CanonicalUser":["*"]},
```

如果按照此方式设置,当referer为"www.example01.com"或"www.example02.com"时,就不能对桶bucket中的资源进行操作。

# 6.4 角色

管理员账户可以设置用户组的角色(Roles),不同角色有不同的权限范围,例如:

#### 表 6-10 角色示例

名称	描述	
OBS Administrator	OBS管理员角色,拥有OBS服务所有OBS操作的权限	
Tenant Guest	访客角色, 仅拥有查询、列举、下载等读操作权限	

对象存储服务的角色在"统一身份认证服务"界面,"管理区域或项目"选项卡的"对象存储服务"子项中设置。

# 6.5 同时使用 ACL 和桶策略

如果ACL和Bucket Policy同时使用,则ACL对某个账户的授权结果allow,可以被Bucket Policy的显示 Deny覆盖。

如果Bucket Policy和IAM Policy同时使用,同样遵循explicit deny>allow>default deny的规则。

SSE-KMS服务端加密对象,不支持Bucket ACL/Policy进行跨租户授权访问。

# 6.6 FLASH 跨域访问配置

OBS系统提供的默认配置为根域名跨域访问允许所有域访问,会使客户端存在被攻击的风险。

客户端可通过在自己的桶内创建配有自己规则的crossdomain.xml,并在其flash代码中添加Security.loadPolicyFile("http://bucket.obs.cn-north-1.myhwclouds.com/crossdomain.xml"),从而避免被攻击的风险。

crossdomain.xml需严格遵守XML语法,有且仅有一个根节点cross-domain-policy,且不包含任何属性。在此根节点下只能包含如下的子节点: site-control、allow-access-from、allow-access-from-identity、allow-http-request-headers-from。下表为对其子节点的介绍。

## 表 6-11

名称	描述
site-control	检查属性值,确认是否可以允许加载其他策略 文件。
	属性值如下:
	none: 不允许使用loadPolicyFile方法加载任何策略文件;
	master-only: 只允许使用主策略文件[默认值];
	by-content-type: 只允许使用loadPolicyFile方法 加载HTTP/HTTPS协议下Content-Type为text/x- cross-domain-policy的文件作为跨域策略文件;
	<b>by-ftp-filename</b> : 只允许使用loadPolicyFile方法 加载FTP协议下文件名为crossdomain.xml的文件 作为跨域策略文件;
	all: 可使用loadPolicyFile方法加载目标域上的任何文件作为跨域策略文件。
allow-access-from	检查属性值,确认能读取本域内容的flash文件 来源域。
	属性值如下:
	domain: 该属性指定一个确切的 IP地址、一个确切的域或一个通配符域(任何域)。只有domain中指定的域,才有权限通过flash读取本域中的内容;
	to-ports: 允许访问读取本域内容的socket连接端口范围;
	secure: 指明信息是否经加密传输。
allow-access-from-identity	为允许有特定证书的来源跨域访问本域上的资源。
allow-http-request-headers-from	授权第三方域将数据以http头的形式发送到本域中。
	属性值如下:
	domain:该属性指定一个确切的 IP地址、一个确切的域或一个通配符域(任何域)。只有domain中指定的域,才有权限通过flash读取本域中的内容;
	headers: 以逗号隔开的列表,表明允许发送的 http头。可用通配符(*)表示全部http头;
	secure: 指明信息是否经加密传输。

# **了** 桶的基础操作

- 7.1 获取桶列表
- 7.2 创建桶
- 7.3 列举桶内对象
- 7.4 获取桶元数据
- 7.5 获取桶区域位置
- 7.6 获取桶存量信息
- 7.7 删除桶

# 7.1 获取桶列表

OBS用户可以通过请求查询自己创建的桶列表。

# 请求消息样式

GET / HTTP/1.1

Host: obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

该请求消息中不带请求参数。

## 请求消息头

该请求消息头使用公共消息字段,具体请参见公共请求消息头章节。

# 请求消息元素

该请求消息中不带有请求参数。

## 响应消息样式

HTTP/1.1 status\_code
Content-Type: type

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应消息中,会以XML形式将用户拥有的桶列出来,元素的具体含义如表7-1所示。

#### 表 7-1 响应消息元素

元素名称	描述
ListAllMyBucketsResult	用户的桶列表。
	类型: XML。
Owner	用户信息,包含用户DomainId和用户 名。
	类型: XML。
ID	用户的DomainId。
	类型:字符串。
DisplayName	用户名。
	类型:字符串。
Buckets	用户所拥有的桶列表。
	类型: XML。
Bucket	具体的桶信息。
	类型: XML。
Name	桶名称。
	类型:字符串。
CreationDate	桶的创建时间。
	类型:字符串。

## 错误响应消息

该请求无特殊错误,所有错误已经包含在表4-1中。

# 7.2 创建桶

创建桶是指按照用户指定的桶名创建一个新桶的操作。

#### □ 说明

一个用户可以拥有的桶的数量不能超过100个。

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成功。除此以外的其他场景重复创建同名桶返回桶已存在。用户可以在请求消息头中加入x-amz-acl参数,设置要创建桶的权限控制策略。

注:

用户删除桶后,需要等待30分钟才能在其他区域创建同名桶。

## 存储类型

允许用户创建不同默认存储类型的桶。发送创桶请求时携带头域"x-default-storage-class"来指定桶的默认存储类型。桶内对象的存储类型与桶默认存储类型保持一致。存储类型有3种:STANDARD(标准存储)、STANDARD\_IA(低频访问存储)、GLACIER(归档存储)。如果没有携带此头域,则创建的桶为标准存储类型。

当往桶内上传对象时,如果没有指定对象的存储类别(参考**10.1 PUT上传**),则该对象的存储类型取桶的默认存储类型。

- OBS标准存储拥有低访问时延和较高的吞吐量,因而适用于有大量热点文件需要 频繁访问数据的业务场景,例如:大数据、移动应用、热点视频、社交图片等场景。
- OBS低频访问存储适用于不频繁访问(少于每月一次访问)但在需要时也要求快速访问数据的业务场景,例如:文件同步/共享、企业备份等场景。与标准存储相比,低频访问存储有相同的数据持久性、吞吐量以及访问时延,且成本较低,但是可用性略低于标准存储。
- OBS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,例如:数据归档、长期备份等场景。归档存储安全、持久且成本极低,可以用来替代磁带库。为了保持成本低廉,数据取回时间可能长达数分钟到数小时不等。

# 请求消息样式

PUT / HTTP/1.1

Host: bucketname. obs. cn-north-1. myhwclouds. com

Content-Length: *length* 

Date: date

Authorization: authorization

 $$$ \cond Create Bucket Configuration xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/"> \cond Constraint>location Constraint> (Create Bucket Configuration)$ 

## 请求消息参数

该请求消息中不带请求参数。

# 请求消息头

该操作消息头与普通请求一样,请参见**公共请求消息头**章节,但可以带附加消息头,附加请求消息头如**表7-2**所示。

表 7-2 附加请求消息头

消息头名称	描述	是否必选
x-amz-acl	创建桶时,可以加上此消息头设置桶的权限控制策略,使用的策略为预定义的常用策略,包括: private、public-read、public-read-write、authenticated-read、bucket-owner-read、bucket-owner-full-control(各策略详细说明见表6-3)。类型:字符串。	否
x-default-storage- class	创建桶时,可以加上此消息头设置桶的默认存储类型,默认存储类型有3种: STANDARD(标准存储)、STANDARD_IA(低频访问存储)、GLACIER(归档存储)。如果没有指定此头域,则创建的桶为标准存储类型。类型:字符串。	沿
x-amz-grant-read	允许列举对象。 类型: 字符串 示例:x-amz-grant-read:id=账户id	否
x-amz-grant-write	允许创建,删除,覆盖桶内所有对象。 类型: 字符串 示例:x-amz-grant-write:id=账户id	否
x-amz-grant-read- acp	允许读桶的ACL信息。 类型: 字符串 示例:x-amz-grant-read-acp:id=账户id	否
x-amz-grant-write- acp	允许修改桶的ACL信息。 类型: 字符串 示例:x-amz-grant-write-acp:id=账户id	否
x-amz-grant-full- control	得到全部权限。 类型: 字符串 示例:x-amz-grant-full-control:id=账户id	否

# 请求消息元素

该操作可以带附加请求消息元素,附加请求消息元素的具体描述如**公共请求消息头**所示。

#### 表 7-3 附加请求消息元素

元素名称	描述	是否必选
LocationConstraint	指定Bucket在哪个区域被创 建。	可选
	如果使用的终端节点归属于默 认区域华北-北京1(cn- north-1)可以不携带此参数; 如果使用的终端节点归属于其 他区域,则必须携带此参数。 当前有效的区域名称有: cn- north-1、cn-east-2、cn- south-1。	
	有关OBS区域和终端节点的更 多信息,请参考 <b>地区和终端节</b> 点。	
	格式:字符串。	

## 响应消息样式

HTTP/1.1 status\_code Location: location Date: date

Content-Length: length

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

## 响应消息元素

该请求的响应中不带有响应元素。

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 7.3 列举桶内对象

对桶拥有读权限的用户可以执行获取桶内对象列表的操作。

如果用户在请求的URI里只指定了桶名,即GET/BucketName,则返回信息中会包含桶内部分或所有对象的描述信息(一次最多返回1000个对象信息);如果用户还指定了prefix、marker、max-keys、delimiter参数中的一个或多个,则返回的对象列表将按照如表7-4所示规定的语义返回指定的对象。

用户也可以请求参数中添加versions参数来执行列举桶内多版本对象的操作。

# 请求消息样式

GET /? HTTP/1.1

Host: bucketname. obs. cn-north-1. myhwclouds. com

Date: date

Authorization: authorization

# 请求消息样式(多版本)

GET /?versions HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

## 请求消息参数

该请求可以通过带参数,列举出桶内的一部分对象,参数的具体含义如表7-4所示。

#### 表 7-4 请求消息参数

参数名称	描述	是否必选
prefix	列举以指定的字符串prefix开头的对象。 类型:字符串。	否
marker	列举桶内对象列表时,指定一个标识符,返回的对象列 表将是按照字典顺序排序后这个标识符以后的所有对 象。 类型:字符串。	否
max-keys	指定返回的最大对象数,返回的对象列表将是按照字典顺序的最多前max-keys个对象,范围是[1,1000],超出范围时,按照默认的1000进行处理。 类型:整型。	否
delimiter	用来分组桶内对象的字符串。所有名字包含指定的前缀且第一次出现的Object作为一组元素CommonPrefixes。 类型:字符串。	否
version- id-marker	与响应中的key-marker配合使用,返回的对象列表将是按照字典顺序排序后在该标识符以后的所有对象。如果version-id-marker不是key-marker的一个版本号,则该参数无效。 类型:字符串 有效值:对象的版本号 约束:不能为空字符串	否

# 请求消息头

该请求使用公共的请求消息头,具体如公共请求消息头章节所示。

# 请求消息元素

该请求消息头中不带消息元素。

# 响应消息样式

HTTP/1.1 status\_code

Date: date

x-amz-bucket-region: region Content-Type: application/xml Content-Length: length

<Response Body>

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应消息中,会以XML形式将桶中的对象列出来,元素的具体含义如表7-5所示。

#### 表 7-5 响应消息元素

元素名称	描述
ListBucketResult	桶中对象列表。
	类型: XML
Contents	对象的元数据信息。
	类型: XML
	父节点: ListBucketResult
CommonPrefixes	请求中带delimiter参数时,返回消息带
	CommonPrefixes分组信息。
	类型: XML
	父节点: ListBucketResult
Delimiter	请求中携带的delimiter参数。
	类型:字符串
	父节点: ListBucketResult
DisplayName	对象所有者的名字。
	类型:字符串
	父节点: ListBucketResult.Contents.Owner
ETag	对象的MD5值。(当对象是服务端加密的 对象时,ETag值不是对象的MD5值。)
	类型:字符串
	父节点: ListBucketResult.Contents
ID	对象所有者的DomainId。
	类型:字符串
	父节点: ListBucketResult.Contents.Owner

元素名称	描述
IsTruncated	表明是否本次返回的ListBucketResult结果列表被截断。"true"表示本次没有返回全部结果;"false"表示本次已经返回了全部结果。 类型:Boolean 父节点:ListBucketResult
Key	对象名。 类型:字符串 父节点:ListBucketResult.Contents
LastModified	对象最近一次被修改的时间。 类型: Date 父节点: ListBucketResult.Contents
Marker	列举对象时的起始位置。 类型:字符串 父节点:ListBucketResult
NextMarker	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等于该值来列举剩余的对象。类型:字符串 父节点:ListBucketResult
MaxKeys	列举时最多返回的对象个数。 类型:字符串 父节点:ListBucketResult
Name	本次请求的桶名。 类型:字符串 父节点:ListBucketResult
Owner	用户信息,包含用户DomainId和用户名。 类型: XML 父节点: ListBucketResult.Contents
Prefix	对象名的前缀,表示本次请求只列举对象名能匹配该前缀的所有对象。类型:字符串 父节点: ListBucketResult
Size	对象的字节数。 类型:字符串 父节点:ListBucketResult.Contents

元素名称	描述
StorageClass	对象的存储类型。
	类型: 枚举值
	有效值: STANDARD   STANDARD_IA   GLACIER
	父节点: ListBucketResult.Contents

## 表 7-6 列举多版本对象响应消息元素

元素名称	描述
ListVersionsResult	保存列举桶中对象列表(含多版本)请 求结果的容器。 类型:容器
Name	桶名。 类型:字符串 父节点:ListVersionsResult
Prefix	对象名的前缀,表示本次请求只列举对 象名能匹配该前缀的所有对象。类型: 字符串 父节点: ListVersionsResult
KeyMarker	列举对象时的起始位置。 类型:字符串 父节点:ListVersionsResult
VersionIdMarker	列举对象时的起始位置。 类型:字符串 父节点:ListVersionsResult
NextKeyMarker	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。 类型:字符串 父节点: ListVersionsResult。
NextVersionIdMarker	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。 类型: 字符串 父节点: ListVersionsResult。
MaxKeys	列举时最多返回的对象个数。 类型:字符串 父节点:ListVersionsResult

元素名称	描述
IsTruncated	表明是否本次返回的ListVersionsResult结果列表被截断。"true"表示本次没有返回全部结果;"false"表示本次已经返回了全部结果。
	类型: 布尔值
	父节点: ListVersionsResult
Version	保存版本信息的容器
	类型:容器
	父节点: ListVersionsResult
DeleteMarker	保存删除标记的容器
	类型: 容器
	父节点: ListVersionsResult
Key	对象名。
	类型: 字符串
	父节点: ListVersionsResult.Version
	ListVersionsResult.DeleteMarker
VersionId	对象的版本号。
	类型:字符串
	父节点: ListVersionsResult.Version   ListVersionsResult.DeleteMarker
IsLatest	标识对象是否是最新的版本,true代表是 最新的版本。
	类型: 布尔值
	父节点: ListVersionsResult.Version   ListVersionsResult.DeleteMarker
LastModified	对象最近一次被修改的时间。
	类型: Date
	父节点: ListVersionsResult.Version   ListVersionsResult.DeleteMarker
ЕТад	对象的MD5值。
	类型:字符串
	父节点: ListVersionsResult.Version
Size	对象的字节数。
	类型:字符串
	父节点: ListVersionsResult.Version

元素名称	描述
Owner	用户信息,包含用户DomainId和用户 名。
	类型: 容器
	父节点: ListVersionsResult.Version   ListVersionsResult.DeleteMarker
ID	对象所有者的DomainId。
	类型:字符串
	父节点: ListVersionsResult.Version.Owner   ListVersionsResult.DeleteMarker.Owner
DisplayName	对象所有者的名字。
	类型:字符串
	父节点: ListVersionsResult.Version.Owner   ListVersionsResult.Version.Owner
StorageClass	对象的存储类型。
	类型: 枚举值
	父节点: ListVersionsResult.Version
CommonPrefixes	请求中带delimiter参数时,返回消息带 CommonPrefixes分组信息。
	类型: 容器
	父节点: ListVersionsResult。
Prefix	CommonPrefixes分组信息中,表明不同的Prefix。
	类型:字符串
	父节点: ListVersionsResult.CommonPrefixes。

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 7.4 获取桶元数据

对桶拥有读权限的用户可以执行查询桶元数据是否存在的操作。

# 请求消息样式

HEAD / HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: *date* 

Authorization: authorization

## 请求消息参数

该请求消息中不带消息参数。

## 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

## 请求消息元素

该请求消息中不带消息元素。

## 响应消息样式

HTTP/1.1 status\_code x-amz-bucket-region: region

Date: date

## 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

除公共响应消息头之外,还可能使用如下表7-7中的消息头。

#### 表 7-7 附加响应消息头

消息头名称	描述
x-amz-bucket-region	桶的区域位置信息。 类型:字符串。
x-default-storage-class	此头域表明当前桶的默认存储类型。包含: STANDARD(标准存储)、STANDARD_IA (低频访问存储)、GLACIER(归档存储)。 类型:字符串。

# 响应消息元素

该请求的响应中不带有响应元素。

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 7.5 获取桶区域位置

对桶拥有读权限的用户可以执行获取桶区域位置信息的操作。

## 请求消息样式

GET /?location HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

## 请求消息参数

该请求消息中不带消息参数。

# 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

## 请求消息元素

该请求消息中不带消息元素。

## 响应消息样式

HTTP/1.1 status\_code

Date: date Content-Type: type Content-Length: length

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<CreateBucketConfiguration xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">

 $\verb|\LocationConstraint|| Location \verb|\Constraint|| Location \| Location$ 

</CreateBucketConfiguration>

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该响应中将桶的区域信息以消息元素的形式返回,元素的具体含义如<mark>公共响应消息头</mark> 所示。

#### 表 7-8 响应消息元素

元素名称	描述	
CreateBucketConfiguration	保存桶区域位置请求结果,包含桶的区域位置信息。 类型:XML。	
LocationConstraint	桶的区域位置信息。 类型:字符串。	

## 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 7.6 获取桶存量信息

桶的拥有者可以执行获取桶存量信息的操作,返回信息包含指定桶的空间大小以及对象个数,桶的空间大小值为非负整数,单位为Byte(字节)。

## 请求消息样式

GET /?storageinfo HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

## 请求消息参数

该请求不使用请求消息参数。

# 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

## 请求消息元素

该请求消息中不使用请求消息元素。

# 响应消息样式

HTTP/1.1 status\_code

Date: date

Content-Type: type Content-Length: length

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

 $\label{lem:composition} $$ \end{composition} $$ GetBucketStorageInfoResult $$ xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/"> $$ \end{composition} $$ \end{compos$ 

⟨Size⟩*size*⟨/Size⟩

<0bjectNumber>number

</ GetBucketStorageInfoResult>

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该响应中将桶存量信息以消息元素的形式返回,元素的具体含义如**表 1 响应消息元素** 所示。

#### 表 7-9 响应消息元素

元素名称	描述
GetBucketStorageInfoResult	保存桶存量请求结果,包含存量大小和 对象个数。
	类型: XML。

元素名称	描述
Size	返回存量大小。 类型:整型。
ObjectNumber	返回对象个数。 类型:整型。

## 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 7.7 删除桶

删除桶操作用于删除用户指定的桶。只有桶的所有者或者拥有桶的删桶policy权限的用户可以执行删除桶的操作,要删除的桶必须是空桶。如果桶中有对象或者有多段任务则认为桶不为空,可以使用列举桶内对象和列举出多段上传任务接口来确认桶是否为空。

注:

如果删除桶时,服务端返回5XX错误或超时,系统需要10分钟左右的时间进行桶信息一致性处理,在此期间桶的信息会不准确。

# 请求消息样式

DELETE / HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: *authorization* 

# 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用公共的请求消息头,具体请参见公共请求消息头章节。

# 请求消息元素

该请求消息中不使用消息元素。

# 响应消息样式

HTTP/1.1 status\_code

Date: date

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应消息中不带消息元素。

# 错误响应消息

无特殊错误,错误已经包含在表4-1中。

# 8桶策略配置

- 8.1 设置桶策略
- 8.2 设置桶配额
- 8.3 设置桶ACL
- 8.4 设置桶日志管理配置
- 8.5 设置桶的生命周期配置
- 8.6 设置桶的多版本状态
- 8.7 设置桶的消息通知配置
- 8.8 设置桶默认存储类型
- 8.9 设置桶的跨区域复制配置
- 8.10 获取桶策略
- 8.11 获取桶配额
- 8.12 获取桶ACL
- 8.13 获取桶日志管理配置
- 8.14 获取桶的生命周期配置
- 8.15 获取桶的多版本状态
- 8.16 获取桶的消息通知配置
- 8.17 获取桶默认存储类型
- 8.18 获取桶的跨区域复制配置
- 8.19 删除桶策略
- 8.20 删除桶的生命周期配置
- 8.21 删除桶的跨区域复制配置

# 8.1 设置桶策略

该接口的实现使用policy子资源创建或者修改一个桶的策略。如果桶已经存在一个策略,那么当前请求中的策略将完全覆盖桶中现存的策略。

要使用该接口,使用者要求必须是桶的所有者。

## 请求消息样式

PUT /?policy HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: signatureValue

Policy written in JSON

## 请求消息参数

该请求消息中不使用消息参数。

## 请求消息头

该请求使用公共消息头,具体请参见公共请求消息头章节。

## 请求消息元素

请求消息体是一个符合JSON格式的字符串,包含了桶策略的信息。具体JSON格式的元素介绍,请参考**6.3 桶策略**。

## 响应消息样式

HTTP/1.1 status\_code

Date: date

Content-Length: length

## 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 响应消息元素

该请求的响应消息中不带有响应元素。

#### 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.2 设置桶配额

只有桶的owner才可以修改桶配额值,桶的拥有者的状态是inactive状态不可以更改桶配额信息。桶空间配额值必须为非负整数,单位为Byte(字节),能设的最大值为  $2^{63}$ -1,配额值设为0表示桶的配额没有上限。

## 请求消息样式

PUT /?quota HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Content-Length: *length* Authorization: *authorization* 

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Quota xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">

<StorageQuota>value</StorageQuota>

</Quota>

# 请求消息参数

该请求在请求消息中没有带有参数。

# 请求消息头

该请求没有特殊的请求消息头,公共部分参见公共请求消息头章节。

## 请求消息元素

该操作需要附加请求消息元素来指定桶的空间配额,具体见表8-1。

#### 表 8-1 附加请求消息元素

元素名称	描述	是否必选
StorageQuota	指定桶空间配额值单位为字节。	是
	类型:整型。	

# 响应消息样式

HTTP/1.1 status\_code

Date: date

Content-Length: length

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应中不带有响应元素。

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.3 设置桶 ACL

OBS支持对桶操作进行权限控制。默认情况下,只有桶的创建者才有该桶的读写权限。用户也可以设置其他的访问策略,比如对一个桶可以设置公共访问策略,允许所有人对其都有读权限。

OBS用户在创建桶时可以设置权限控制策略,也可以通过ACL操作API接口对已存在的桶更改或者获取ACL(access control list)。

## 请求消息样式

```
PUT /?acl HTTP/1.1
Host: bucketname.obs.cn-north-1.myhwclouds.com
Date: date
Authorization: authorization
Content-Type: application/xml
Content-Length: length
<AccessControlPolicy>
   <0wner>
       <ID>ID</ID>
       <DisplayName>displayname
   \langle AccessControlList \rangle
       <Grant>
           <Grantee>grantee
           <Permission>permission
   </AccessControlList>
</AccessControlPolicy>
```

# 请求消息参数

该操作请求不带消息参数。

# 请求消息头

使用者可以使用canned ACL的方式来更改桶的ACL,即通过头域(x-amz-acl)的方式设置ACL。每一种canned ACL都有一套自己预先定义好的被授权用户以及相应权限,通过canned ACL的方式授予访问权限,使用者必须添加以下的头域并且指定canned ACL的值。

表 8-2 头域方式设置桶 ACL

名称	描述	是否必须
x-amz-acl	通过canned ACL的方式来 设置桶的ACL。 类型:字符串 有效值:private   public- read   public-read-write   authenticated-read   bucket- owner-read   bucket-owner- full-control   log-delivery- write	否

## 请求消息元素

更改桶的ACL请求需要在消息元素中带上ACL信息,元素的具体含义如**公共请求消息** 头所示。

#### 表 8-3 附加请求消息元素

元素名称	描述	是否必选
Owner	桶的所有者信息,包含ID和 DisplayName两个元素。 类型:XML。	否
ID	被授权用户的DomainId。 类型:字符串。	否
DisplayName	被授权用户的用户名 类型:字符串。	否
Grant	用于标记用户及用户的权限。 类型:XML。	否
Grantee	记录用户信息。详情参考 <b>表6-1</b> 类型: XML。	否
Permission	授予的权限。 <b>表6-2</b> 类型:枚举类型。	否
AccessControlList	访问控制列表,包含Grant、Grantee、Permission三个元素。 类型:XML。	否

## 响应消息样式

HTTP/1.1 status\_code

Date: date

Content-Length: length

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应消息中不带有响应元素。

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.4 设置桶日志管理配置

提供日志访问的设置功能,通过这个设置可以指定哪些用户可以查看和修改桶的日志文件参数配置。利用put方法并加入logging子资源来完成桶的日志配置参数的设置。在当前版本下,只有桶的所有者或者被桶策略授权s3:PutBucketLogging权限的用户才有对桶日志配置参数查看和修改的权限。

创建桶时,默认是不生成桶的日志的,如果需要生成桶的日志,该桶需要打开日志配置管理的开关,同时授予日志投递用户组对存放日志桶的WRITE和READ\_ACP权限。配置管理开关的打开方法是在上传的logging文件中在LoggingEnabled标签下配置相应的日志管理功能。对应的,关闭的方法是上传一个带有空的BucketLoggingStatus标签的logging文件。

在设置桶日志管理配置时,源桶的所有者可以指定一个目标桶,将生成的所有日志放到该桶中。源桶所有者只能将日志投递到其所拥有的任意一个桶,同时目标桶必须是和源桶在同一个区域(Region)内的桶。此外,默认存储类别为低频访问存储或归档存储的桶不能作为目标桶。

## 请求消息样式

PUT /?logging HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: *date* 

Authorization: signatureValue

logging configuration

# 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用公共消息头,具体请参见公共请求消息头章节。

# 请求消息元素

#### 表 8-4 请求消息元素表

名字	描述	是否必选
BucketLoggingStatus	日志状态信息的容器。 类型:容器	是
LoggingEnabled	该元素起到对日志配置管理的使能作用(呈现此元素则打开日志配置,否则则关闭配置)。在此元素下,可加入具体的日志配置信息。 类型:容器	否

名字	描述	是否必选
Grant	是被授权者及其权限的容器。 类型:容器	否
Grantee	作为被授权logging权限用户的容器。 类型:容器	否
ID	被授权者的DomainId,全 局唯一标识。 类型:字符串	否
DisplayName	被授权用户的名称,并非 全局唯一,但是一个用户 ID只会绑定一个用户名。 类型:字符串	否
Permission	对于一个桶的日志访问权限来说,owner在创桶时将自动获得对源桶的FULL_CONTROL权限。不同的权限决定了对不同日志的访问限制。类型:字符串权限有效值:FULL_CONTROL READ WRITE	否
TargetBucket	在生成日志时,源桶的所有者将生成的所有日志的所有目志的所有自志的所有有相。源子的人。 为了将生成的所有有相的的一个。 为了将生人。 为了, 为一个。 为了, 为一个。 为一个。 为一个。 为一个。 为一个。 为一个。 为一个。 为一个。	是

名字	描述	是否必选
TargetPrefix	通过该元素指定一个前缀,所有生成的日志对象的对象名都以此元素的内容为前缀。 类型:字符串	是
TargetGrants	授权信息的容器。 类型:容器	否

## 响应消息样式

HTTP/1.1 status\_code

Date: date

Content-Length: length

## 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

## 响应消息元素

该请求的响应消息中不带有响应元素。

## 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.5 设置桶的生命周期配置

OBS系统支持指定规则来实现定时删除桶中对象,这就是生命周期配置。典型的应用场景如:

- 周期性上传的日志文件,可能只需要保留一个星期或一个月,到期后要删除它们。
- 某些文档在一段时间内经常访问,但是超过一定时间后就可能不会再访问了。这 种文档您可能会先选择归档,然后在一定时间后删除。
- 桶的最低转温和转冷的时间可以配置,取值范围为24~8640。

本接口实现为桶创建或更新生命周期配置信息。

要正确执行此操作,需要确保执行者有s3:PutLifecycleConfiguration权限。默认情况下只有桶的所有者可以执行此操作,也可以通过设置桶策略或用户策略授权给其他用户。

生命周期配置实现了定时删除对象和定时迁移对象的功能,所以如果想要阻止用户删除或迁移对象,以下几项操作的权限都应该被禁止:

- s3:DeleteObject
- s3:DeleteObjectVersion

#### • s3:PutLifecycleConfiguration

如果想要阻止用户管理桶的生命周期配置,应该禁止s3:PutLifecycleConfiguration权限。

## 请求消息样式

```
PUT /?lifecycle HTTP/1.1
Host: bucketname. obs. cn-north-1. myhwclouds. com
Content-Length: length
Date: date
Authorization: authorization
Content-MD5: MD5
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
    <Rule>
        <ID>id</ID>
        <Prefix>prefix</Prefix>
        <Status>status</Status>
        <Expiration>
             <Days>days
        </Expiration>
        <\!NoncurrentVersionExpiration>
             NoncurrentDays>days/NoncurrentDays>
        </NoncurrentVersionExpiration>
        \langle Transition \rangle
         <Days>30</Days>
           \label{lass} $$\footnotesize \STANDARD_IA < \StorageClass > $$
         </Transition>
        \langle Transition \rangle
          <Days>60</Days>
         <StorageClass>GLACIER</StorageClass>
         </Transition>
         <NoncurrentVersionTransition>
          <NoncurrentDays>30</NoncurrentDays>
          \verb| \langle StorageClass \rangle STANDARD_IA < / StorageClass \rangle|
         </NoncurrentVersionTransition>
         <NoncurrentVersionTransition>
          <NoncurrentDays>60</NoncurrentDays>
          <StorageClass>GLACIER</StorageClass>
        </NoncurrentVersionTransition>
    </Rule>
</LifecycleConfiguration>
```

# 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用的消息头如下表8-5所示。

#### 表 8-5 请求消息头

消息头名称	描述	是否必选
Content-MD5	按照RFC 1864标准计算出消息体的MD5摘要字符串,即消息体128-bit MD5值经过base64编码后得到的字符串。	是
	类型: 字符串	
	示例: n58IG6hfM7vqI4K0vnWpog==	

## 请求消息元素

在此请求中,需要在请求的消息体中配置桶的生命周期配置信息。配置信息以XML格式上传,具体的配置元素如表8-6描述。

如果桶的多版本是Enabled或者Suspended,那么可以设置NoncurrentVersionTransition或NoncurrentVersionExpiration来控制对象的历史版本的生命周期。一个历史版本的生命周期,取决于它成为历史版本的时刻(即被新版本覆盖的那个时刻)和NoncurrentDays。对于删除来说,例如NoncurrentDays配置为1的话,表示当一个版本成为历史版本之后,再过1天才能删除。对象A的版本V1创建于1号,5号的时候又上传新的版本V2,此时V1成为历史版本,那么再过1天,7号的0点,V1就过期了。如果该版本不满足删除,迁移配置NoncurrentDays为1,StorageClass为STANDARD\_IA的话,表示当一个版本成为历史版本之后,再过1天转为温对象。对象A的版本V1创建于1号,5号的时候又上传新的版本V2,此时V1成为历史版本,那么再过1天,7号的0点,V1就会迁移成温对象了。(备注:对象过期后被删除的时间或对象迁移的时间可能会有一定的延迟,一般不超过48小时。)

如果桶的多版本是Enabled或者Suspended,且最新版本对象满足Expiration规则时的处理:

● 桶当前的多版本状态为enabled:

如果对象的最新版本不是deletemarker,则该对象会产生一个新的deletemarker; 如果最新版本是deletemarker,且该对象只有这一个版本,则这个版本会被删除; 如果最新版本是deletemarker,且对象还有其他版本,则该对象的所有版本维持不 变,没有新增和删除,也不会被修改(即无任何变化)。

● 桶当前的多版本状态为suspended:

如果最新版本不是null版本,则会产生一个新的null版本的deletemarker; 如果最新版本是null版本,则这个null版本会被新产生的null版本的deletemarker覆 盖。

如果桶的多版本是Enabled或者Suspended,且最新版本对象满足Transition规则时的处理:

如果对象的最新版本是deletemarker,则这个版本不被迁移;

如果最新版本不是deletemarker,且该对象满足迁移条件,则这个版本会被迁移。

## 表 8-6 生命周期配置元素

名称	描述	是否必选
Date	表示规则生效的时间。	如果没有Days
	该值必须兼容ISO8601格式,而且必须是 UTC午夜0点。	元素,则必选
	类型:字符串	
	父节点: Expiration, Transition	
Days	表示在对象创建时间后第几天时规则生效 (仅针对对象的最新版本)。	如果没有Date 元素,则必选
	类型: 正整数	
	父节点: Expiration, Transition	
StorageClass	表示对象将被修改成的存储级别。 类型:字符串STANDARD_IA或GLACIER 父节点:Transition, NoncurrentVersionTransition	如果有 Transition或 NoncurrentVersi onTransition元 素,则必选
Transition	生命周期配置中表示迁移时间和迁移后对象存储级别的元素(仅针对对象的最新版本) 类型: XML 子节点: Date或Days,StorageClass 父节点: Rule	如果没有 NoncurrentVersi onTransition, Expiration, NoncurrentVersi onExpiration, 则必选
Expiration	生命周期配置中表示过期时间的Container(仅针对对象的最新版本)。 类型: XML 子节点: Date或Days 父节点: Rule	如果没有 Transition, NoncurrentVersi onTransition, NoncurrentVersi onExpiration,则 必选
ID	一条Rule的标识,由不超过255个字符的字符串组成。 类型:字符串 父节点:Rule	否
LifecycleConfiguratio n	生命周期配置Rule的Container。可以配置多条Rule,但需保证整个配置消息体总大小不超过20KB。 类型: XML 子节点: Rule 父节点: 无	是

名称	描述	是否必选
NoncurrentDays	表示对象在成为历史版本之后第几天时规则生效(仅针对历史版本)。 类型:正整数 父节点:NoncurrentVersionExpiration, NoncurrentVersionTransition	如果有 NoncurrentVersi onExpiration或 NoncurrentVersi onTransition元 素,则必选
NoncurrentVersionTra nsition	生命周期配置中表示对象的历史版本迁移时间和迁移后对象存储级别的元素。 类型: XML 子节点: NoncurrentDays, StorageClass 父节点: Rule	如果没有 Transition, Expiration, NoncurrentVersi onExpiration, 则必选
NoncurrentVersionExp iration	生命周期配置中表示历史版本过期时间的 Container。您可以将该动作设置在已启用 多版本(或暂停)的桶,来让系统删除对 象的满足特定生命周期的历史版本(仅针 对历史版本)。 类型: XML 子节点: NoncurrentDays 父节点: Rule	否
Prefix	对象名前缀,用以标识哪些对象可以匹配到当前这条Rule。 类型:字符串 父节点:Rule	是
Rule	具体某一条生命周期配置的Container。 类型:容器 父节点: LifecycleConfiguration	是
Status	标识当前这条Rule是否启用。 类型:字符串 父节点:Rule 有效值:Enabled,Disabled	是

# 响应消息样式

HTTP/1.1 status\_code

Date: date

Content-Length: length

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

## 响应消息元素

该请求的响应消息不带消息元素。

## 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.6 设置桶的多版本状态

多版本功能可在用户意外覆盖或删除对象的情况下提供一种恢复手段。用户可以使用 多版本功能来保存、检索和还原对象的各个版本,这样用户能够从意外操作或应用程 序故障中轻松恢复数据。多版本功能还可用于数据保留和存档。

默认情况下,桶没有设置多版本功能。

本接口设置桶的多版本状态,用来开启或暂停桶的多版本功能。

设置桶的多版本状态为Enabled,开启桶的多版本功能:

- 上传对象时,系统为每一个对象创建一个唯一版本号,上传同名的对象将不再覆盖旧的对象,而是创建新的不同版本号的同名对象;
- 可以指定版本号下载对象,不指定版本号默认下载最新对象;
- 刪除对象时可以指定版本号删除,不带版本号删除对象仅产生一个带唯一版本号的删除标记,并不删除对象:
- 列出桶内对象列表时默认列出最新对象列表,可以指定列出桶内所有版本对象列表:
- 除了删除标记外,每个版本的对象存储均需计费(不包括对象元数据)。

设置桶的多版本状态为Suspended, 暂停桶的多版本功能:

- 旧的版本数据继续保留;
- 上传对象时创建对象的版本号为null,上传同名的对象将覆盖原有同名的版本号为null的对象;
- 可以指定版本号下载对象,不指定版本号默认下载最新对象:
- 删除对象时可以指定版本号删除,不带版本号删除对象将产生一个版本号为null的 删除标记,并删除版本号为null的对象;
- 除了删除标记外,每个版本的对象存储均需计费(不包括对象元数据)。

只有桶的所有者可以设置桶的多版本状态。

# 请求消息样式

PUT /?versioning HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization Content-Length: length

## 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

## 请求消息元素

在此请求中,需要在请求的消息体中配置桶的多版本状态,配置信息以XML格式上传。具体的配置元素如表8-7描述。

#### 表 8-7 桶的多版本状态配置元素

名称	描述	是否必选
VersioningConfiguration	多版本配置的根节点。	是
	父节点: 无	
Status	标识桶的多版本状态。	是
	类型: 枚举值	
	父节点: VersioningConfiguration	
	有效值: Enabled, Suspended	

## 响应消息样式

HTTP/1.1 status code

Date: date

Content-Length: length

## 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应消息中不带消息元素。

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.7 设置桶的消息通知配置

OBS 消息通知功能能够帮助您对桶的重要的操作及时通知到您,确保您安全、及时知 道发生在桶上的关键事件。有关消息通知的配置和对接开发,需要参考对象存储服务 开发指南中的**消息通知管理**功能章节。

默认情况下,您的桶没有配置事件通知。这个时候桶的通知配置将是一个空 NotificationConfiguration。对已配置有事件通知的桶,可以通过添加空 NotificationConfiguration元素禁用消息通知功能。

```
<NotificationConfiguration>
</NotificationConfiguration>
```

目前对象存储服务(OBS)支持包括简单通知服务(SMN)、函数服务(FunctionStage)以及函数工作流服务(FunctionGraph)在内的三种桶通知服务配置。以SMN为例,当 OBS 接收到配置消息通知的请求后,会验证指定的**消息通知服务(SMN)主题**是否存在及主题策略是否授权给了对象存储服务,验证通过后会向该主题订阅者发送一个测试消息通知。

为了能成功执行此配置操作,需要确保执行者拥有s3:PutBucketNotification权限。默认情况下只有桶的所有者拥有该权限,但可以通过设置桶策略授权给其他用户。

#### □□说明

- 1. 主题策略授权给对象存储服务的操作请参考SMN服务中修改主题策略章节描述。
- 函数服务策略授权给对象存储服务的操作请参考函数服务(FunctionStage)中使用OBS触发器章节描述。
- 3. 函数工作流服务策略授权给对象存储服务的操作请参考函数工作流服务(FunctionGraph)中 创建工作流触发器章节描述。

## 请求消息样式

```
PUT /?notification HTTP/1.1
Host: bucketname.obs.cn-north-1.myhwclouds.com
Date: date
Authorization: authorization string
 <NotificationConfiguration>
                <TopicConfiguration>
                                <Id>ConfigurationId</Id>
                               <Filter>
                                               <S3Key>
                                                              <FilterRule>
                                                                             <Name>prefix</Name>
                                                                              <Value>prefix-value</Value>
                                                              </FilterRule>
                                                              <FilterRule>
                                                                              <Name>suffix</Name>
                                                                              <Value>suffix-value</Value>
                                                              </FilterRule>
                                          </S3Key>
                               </Filter>
                               <Topic>TopicARN</Topic>
                               <Event>event-type</Event>
                               <Event>event-type</Event>
               </TopicConfiguration>
               <CloudFunctionConfiguration>
                               <Id>ConfigurationId</Id>
                               <Filter>
                               </Filter>
                               < CloudFunction > cloud-function-arm < / CloudFunction > cloud-function > cloudFunction > cl
                               <Event>event-type</Event>
                               </CloudFunctionConfiguration>
                <FunctionGraphConfiguration>
                              <Id>ConfigurationId</Id>
                               <Filter>
```

# 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用公共消息头,具体参见章节3公共消息头。

# 请求消息元素

在此请求中,需要在请求的消息体中配置桶的通知,通知的配置信息以XML格式上传。具体的配置元素如表8-8描述。

#### 表 8-8 桶的通知服务配置元素

名称	描述	是否必选
NotificationConfiguration	配置桶事件通知的根元素,如果子元素为 空,说明消息通知功能处于关闭状态。	是
	类型:元素	
	父元素: 无	
	子元素: 0个或多个TopicConfiguration	
TopicConfiguration	配置事件通知主题的元素。	否
	类型:元素	
	父元素: NotificationConfiguration	
	子元素: Id, Filter, Topic, 一个或多个 Event	
Topic	事件通知主题的URN,当OBS检测到桶中 发生特定的事件后,将会发布通知消息至	若是父元 素
	该主题,可以在 <b>消息通知服务主题</b> 部分找 到具体值。	TopicConfi guration添
	类型: 字符串	加后,本
	父元素: TopicConfiguration	元素是必 选项
CloudFunctionConfiguratio	配置函数服务(FunctionStage)的元素。	否
n	类型:元素	
	父元素: NotificationConfiguration	
	子元素: Id, Filter, Topic, 一个或多个	
	Event	

名称	描述	是否必选
CloudFunction	函数服务的URN,当OBS检测到桶中发生特定的事件后,将会发送消息至函数服务调用执行该函数。 类型:字符串 父元素:CloudFunctionConfiguration	若是 素 CloudFunct ionConfigur ation添加 后,本元 素 项
FunctionGraphConfiguratio n	配置函数工作流服务(FunctionGraph)的元素。 类型:元素 父元素:NotificationConfiguration 子元素:Id,Filter,Topic,一个或多个 Event	否
FunctionGraph	函数工作流服务的URN,当OBS检测到桶中发生特定的事件后,将会发送消息至函数工作流服务调用执行该工作流。 类型:字符串 父元素:FunctionGraphConfiguration	若添加父 元素 FunctionGr aphConfigu ration,本 元素是必 选项
Id	每项事件通知配置的唯一标识,若是用户未指定ID,系统将自动分配一个ID。 类型:字符串 父元素:TopicConfiguration, CloudFunctionConfiguration 或者 FunctionGraphConfiguration	否
Filter	S3Key的元素,用以保存过滤对象名的一组规则。 类型:元素 父元素:TopicConfiguration, CloudFunctionConfiguration 或者 FunctionGraphConfiguration 子元素:S3Key	否
S3Key	定义过滤规则的元素,该规则用以匹配对象名前缀和后缀。 类型:元素 父元素:Filter 子元素:一个或者多个FilterRule	否

名称	描述	是否必选
FilterRule	定义过滤规则键值对的元素。 类型:元素 父元素:S3Key 子元素:Name,Value	否
Name	指定规则按对象名前缀或后缀进行过滤。 类型:字符串 父元素:FilterRule 合法值:prefix或者suffix	否
Value	指定的对象名关键字,根据Name元素定义的前缀或后缀,输入需要过滤的对象的关键字信息,字符越长匹配精度越高,最大可支持1024个字符。 类型:字符串 父元素:FilterRule	否
Event	需要发布通知消息的事件类型。 说明 在一个TopicConfiguration、 CloudFunctionConfiguration或者 FunctionGraphConfiguration配置项中可以添加 多个事件类型。 类型:字符串 合法值: s3:ObjectCreated:* s3:ObjectCreated:Put s3:ObjectCreated:Post s3:ObjectCreated:Copy s3:ObjectCreated:CompleteMultipartUpload s3:ObjectRemoved:* s3:ObjectRemoved:Delete s3:ObjectRemoved:Delete s3:ObjectRemoved:DeleteMarkerCreated 父元素:TopicConfiguration, CloudFunctionConfiguration 或者 FunctionGraphConfiguration	若是父元素 TopicConfiguration, CloudFunct ionConfiguration或者 FunctionGraphConfiguration添加后,未必选项

HTTP/1.1 status\_code

Date: date

Content-Length: *length* Content-Type: *type* 

#### 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 响应消息元素

该请求的响应消息中不带有响应元素。

#### 错误响应消息

当用户执行调用本接口时,系统将会检查NotificationConfiguration元素的有效性,以及配置是否有效。下表中列出本接口的一些常见错误,以及可能原因。

#### 表 8-9 配置桶的通知的错误码列表

错误码	描述	HTTP状态 码
InvalidArgument	该错误可能是由于下列原因导致。  ● 指定了不支持的event。  ● 指定的URN不存在或者填写错误,请确认URN合法。  ● 指定的URN中的区域与桶所在的区域不一致,请确保桶所在的区域与URN中的区域一致。  ● 指定的过滤规则之间存在互相包含或者重叠。	400 Bad Request
AccessDenied	执行者不是桶的所有者,或者还未被授权s3:PutBucketNotification权限	403 Forbidden

# 8.8 设置桶默认存储类型

本接口实现为桶创建或更新桶的默认存储类型配置信息。

要正确执行此操作,需要确保执行者有s3:PutBucketStoragePolicy权限。默认情况下只有桶的所有者可以执行此操作,也可以通过设置桶策略或用户策略授权给其他用户。

设置了桶的默认存储类型之后,若上传对象、复制对象和初始化多段上传任务时未指定对象的存储类型,则该对象的存储类型取桶的默认存储类型。

未配置桶的默认存储类型时,桶的默认存储类型为STANDARD(标准存储)。

# 请求消息样式

PUT /?storagePolicy HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date
Content-Type: type
Content-Length: length
Authorization: authorization

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

#### 请求消息参数

该请求在请求消息中没有带有参数。

#### 请求消息头

该请求没有特殊的请求消息头,公共部分参见公共请求消息头章节。

## 请求消息元素

该操作需要附加请求消息元素来指定桶的默认存储类型,具体见表8-10。

#### 表 8-10 附加请求消息元素

元素名称	描述	是否必选
DefaultStorageClass	指定桶默认存储类型。 类型:字符串,存储类型有3种: STANDARD(标准存储)、 STANDARD_IA(低频访问存储)、 GLACIER(归档存储)。因此这里可配 置的值有:STANDARD、 STANDARD_IA和GLACIER,注意大小 写敏感。	是

#### 响应消息样式

HTTP/1.1 status\_code
Date: date

#### 响应消息头

该请求的响应消息使用公共消息头,具体请参见公共请求消息头章节。

# 响应消息元素

该请求的响应消息中不带消息元素。

#### 错误响应消息

无特殊错误,错误已经包含在表4-1中。

# 8.9 设置桶的跨区域复制配置

跨区域复制是指跨不同区域中的桶自动、异步地复制对象。通过启用跨区域复制,OBS可将新创建的对象、对象更新以及对象删除从一个源桶复制到不同区域中的目标桶,且复制过去的对象称之为"副本"对象。跨区域复制有具体要求,根据对象的创建方式或加密方式确定哪些对象可以或不可以跨区域复制。

设置桶的跨区域复制,需要满足以下要求:

- 1. 要求源桶和目标桶均开启多版本,否则配置跨区域复制配置会失败。如何启用桶的多版本,请参见**8.6 设置桶的多版本状态**。
- 2. 源桶和目标桶必须在不同的区域。
- 3. 如果源桶拥有者与对象拥有者不同,则对象拥有者必须通过对象ACL向源桶拥有者授予READ和READ\_ACP权限,请参见10.4-设置对象ACL。如果目标桶拥有者与源桶拥有者不同,需要目标桶拥有者通过桶策略向源桶拥有者授予ReplicateObject和ReplicateDelete权限,请参见8.1 设置桶策略。
- 4. OBS云服务必须有权代表您将对象从源桶复制到目标桶。源桶拥有者通过创建 IAM委托,选择OBS云服务作为代理人,可以通过以下方式创建OBS云服务委托:

#### ● 创建OBS云服务委托:

● 如果目标桶与源桶的拥有者不一致,那么目标桶的桶策略配置如下:

```
{
  "Effect":"Allow",
  "Principal" : {
    "AWS":[ "arn:aws:iam:: source[z1] bucket ownerID:root"]
},
  "Action":[
    "s3:ReplicateObject",
    "s3:ReplicateDelete"
],
  "Resource":"arn:aws:s3:::destination-bucket/*"
}
```

如何设置桶策略,请参见**8.1 设置桶策略**。设置桶策略后,"代理人"(OBS云服务)就有权限可以读取源桶的对象,也有权限将对象复制到目标桶中。

## 请求消息样式

```
PUT /?replication HTTP/1.1
Host: bucketname.obs.cn-north-1.myhwclouds.com
x-amz-date: date
Content-MD5: content-MD5
Authorization: authorization string
Content-Length: contentlength
<ReplicationConfiguration>
     <Agency>exampleagency</Agency>
     <Rule>
        \langle ID \rangle rule1 \langle /ID \rangle
        <Prefix>key-prefix</Prefix>
        <Status>rule-status</Status>
        (Destination)
            <Bucket>arn:aws:s3:::targetbucketname</Bucket>
            <StorageClass>STANDARD</StorageClass>
        </Destination>
     </Rule>
</ReplicationConfiguration>
```

# 请求参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用的消息头如下所示。

#### 表 8-11 设置桶的复制配置请求消息头

名称	描述	是否必选
Content-MD5	按照RFC 1864标准计算出消息体的MD5摘要字符串,即消息体128-bit MD5值经过base64编码后得到的字符串。	是

# 请求消息元素

在此请求中,需要在请求的消息体中配置桶的复制配置,配置信息以XML格式上传。 具体的配置元素如下描述。

#### 表 8-12 设置桶的复制配置元素

名称	描述	是否必须
ReplicationConfigu ration	复制规则的容器,最多可以达到100条规则,所有的复制配置大小可达到50KB。 类型:容器 子节点:Rule 父节点:无	是
Agency	用户创建的委托名字,最大长度为64字符。 类型:字符串 父节点:ReplicationConfiguration	是
Rule	一条特定复制规则信息的容器。 复制配置必须至少配置1条规则,最多能达到100条规则。 类型:容器 父节点: ReplicationConfiguration	是
ID	规则的特殊辨识符,最大长度为255字符。 类型:字符串 父节点:Rule	否
Status	如果Status为Disabled,这条规则会被忽略。 类型:字符串 父节点:Rule 有效值: Enabled,Disabled	是

名称	描述	是否必须
Prefix	对象名的前缀,适配于一个或者多个对象。 最大前缀长度可达到为1024个字节,一个前 缀不能包含已经存在的另一个前缀。 类型:字符串 父节点:Rule	是
Destination	目标桶信息的容器。 类型:容器 父节点:Rule	是
Bucket	存储被规则标识的对象副本的桶名称。 如果在复制配置中有多条规则,这些规则必 须都要标识同一个桶作为目标桶。 类型:字符串 父节点:Destination	是
StorageClass	复制到目标端的副本对象的存储类型。 类型: 枚举值 父节点:Destination 有效值: STANDARD STANDARD_IA  GLACIER	否

HTTP/1.1 status\_code

Server: OBS

Content-Length: contentlength

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应消息中不带有响应元素。

# 错误响应消息

在此请求的响应中不会返回特殊错误。

# 8.10 获取桶策略

该接口的实现使用policy子资源来将指定桶的策略返回给客户端。

要使用该接口,使用者必须是桶的所有者。

如果指定桶本身的策略不存在,则返回404 "NoSuchBucketPolicy"的错误。

#### 请求消息样式

GET /?policy HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

#### 请求消息参数

该请求消息中不使用消息参数。

#### 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

#### 请求消息元素

该请求消息中不使用消息元素。

#### 响应消息样式

HTTP/1.1 status\_code

Content-Type: application/xml

Date: date

Policy Content

## 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 响应消息元素

响应消息体是一个JSON格式的桶策略字符串。

#### 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.11 获取桶配额

桶的拥有者可以执行获取桶配额信息的操作。桶的拥有者的状态是inactive状态不可以查询桶配额信息。桶空间配额值的单位为Byte(字节),0代表不设上限。

#### 请求消息样式

GET /?quota HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

该请示消息中不带消息参数。

#### 请求消息头

该请求使用公共的请求消息头,具体参见公共请求消息头章节。

#### 请求消息元素

该请求消息不使用消息元素。

#### 响应消息样式

HTTP/1.1 status\_code

Date: date

 ${\tt Content-Type: application/xml}$ 

Content-Length: length

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Quota xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">

<StorageQuota>quota

</Quota>

#### 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 响应消息元素

该响应以消息元素的形式返回桶的配额信息,元素的具体意义如表8-13所示。

#### 表 8-13 响应消息元素

元素名称	描述
Quota	桶的配额,包含配额量元素。 类型:XML。
StorageQuota	桶的配额量。单位字节。 类型:字符串。

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.12 获取桶 ACL

用户执行获取桶ACL的操作,返回信息包含指定桶的权限控制列表信息。用户必须拥有对指定桶读ACP的权限或FULL\_CONTROL权限,才能执行获取桶ACL的操作。

## 请求消息样式

GET /?acl HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

#### 请求消息参数

该请求消息中不使用消息参数。

## 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

#### 请求消息元素

该请求消息中不使用消息元素。

#### 响应消息样式

```
HTTP/1.1 status_code
Date: date
Content-Length: length
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AccessControlPolicy xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">
    <0wner>
        <ID>id</ID>
        <DisplayName>name
    </0wner>
    <AccessControlList>
        <Grant>
            <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
xsi:type="CanonicalUser">
                <ID>id</ID>
                <DisplayName>name</DisplayName>
            </Grantee>
            <Permission>permission</Permission>
        \langle / Grant \rangle
    </AccessControlList>
</AccessControlPolicy>
```

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应中以消息元素的形式返回桶的ACL信息,元素的具体意义如表8-14所示。

#### 表 8-14 响应消息元素

元素	元素说明
Owner	桶的所有者信息。 类型:XML。
ID	用户的DomainId。 类型:字符串。
DisplayName	用户的名称。 类型:字符串。

元素	元素说明
AccessControlList	访问控制列表,记录了对该桶有访问权 限的用户列表和这些用户具有的权限。 类型:XML。
Grant	用于标记用户及用户的权限。 类型:XML。
Grantee	记录用户信息。 类型:XML。
Permission	指定的用户对该桶所具有的操作权限。 类型:字符串。

#### 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.13 获取桶日志管理配置

该接口的目的是查询当前桶的日志管理配置情况。其实现是通过使用http的get方法再加入logging子资源来返回当前桶的日志配置情况。

要使用该接口,使用者必须是桶的所有者或者是被桶策略授权s3:GetBucketLogging权限的用户。

# 请求消息样式

GET /?logging HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

# 请求消息元素

该请求消息中不使用消息元素。

# 响应消息样式

HTTP/1.1 status\_code

Content-Type: application/xml

Date: date

Content-Length: length

```
<?xml version="1.0" encoding="UTF-8"?>
<LoggingEnabled>
   <TargetBucket>bucketName</TargetBucket>
   <TargetPrefix>prefix</TargetPrefix>
      <TargetGrants>
         <Grant>
Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
                <ID>id</ID>
                <DisplayName>displayName
            </Grantee>
            <Permission>permission</Permission>
         </Grant>
      </TargetGrants>
   </BucketLoggingStatus>
```

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

## 响应消息元素

该请求的响应中以消息元素的形式返回桶的日志信息,元素的具体意义如**表1 响应消息** 元**素**所示。

#### 表 8-15 响应消息元素

名字	描述	是否必选
BucketLoggingStatus	logging状态信息的容器。 类型:容器	是
LoggingEnabled	用于logging信息的容器。 并且该元素起到对logging 配置管理的使能作用(呈 现此元素则打开logging配 置,否则则关闭)。 类型:容器	否
Grant	是被授权者及其权限的容器。 类型:容器	否
Grantee	作为被授权logging权限用户的容器。 类型:容器	否
ID	被授权用户的Domain Id, 全局唯一标识。 类型:字符串	否

名字	描述	是否必选
DisplayName	被授权用户的名称,并非全局唯一,但是一个用户ID只会绑定一个用户名。 类型:字符串	否
Permission	对于一个桶的logging权限来说,owner在创桶时将自动获得对源桶的FULL_CONTROL权限。不同的权限决定了对不同日志的访问限制。 类型:字符串权限有效值:FULL_CONTROL READ WRITE	否
TargetBucket	在生成日志时,源桶的 owner可以指定一个目标 桶,将生成的所有日志放 到该桶中。在OBS系统 中,支持多个源桶生成的 日志放在同一个目标桶 中,如果这样做,就需要 指定不同的TargetPrefix以 达到为来自不同源桶的日 志分类的目的。 类型:字符串	否
TargetPrefix	通过该元素可以指定一个 前缀给一类日志生成的对 象。 类型:字符串	否
TargetGrants	授权信息的容器。 类型:容器	否

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.14 获取桶的生命周期配置

获取该桶设置的生命周期配置信息。

要正确执行此操作,需要确保执行者有s3:GetLifecycleConfiguration执行权限。默认情况下只有桶的所有者可以执行此操作,也可以通过设置桶策略或用户策略授权给其他用户。

#### 请求消息样式

```
GET /?lifecycle HTTP/1.1
Host: bucketname.obs.cn-north-1.myhwclouds.com
Date: date
Authorization: authorization
```

#### 请求消息参数

该请求消息中不使用消息参数。

## 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

#### 请求消息元素

该请求消息中不使用消息元素。

#### 响应消息样式

```
HTTP/1.1 status_code
Date: date
Content-Type: application/xml
Date: date
Content-Length: length
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<LifecycleConfiguration xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">
    <Rule>
        \langle {\rm ID} \rangle id \! \langle /{\rm ID} \rangle
        <Prefix>prefix</Prefix>
        <Status>status</Status>
        <Expiration>
             <Date>date
        </Expiration>
        <NoncurrentVersionExpiration>
             <NoncurrentDays>days/NoncurrentDays>
         </NoncurrentVersionExpiration>
        <Transition>
          \label{eq:date} $$\langle Date \rangle date \langle Date \rangle $$
         <StorageClass>STANDARD_IA</StorageClass>
        </Transition>
         <Transition>
          <Date>date</Date>
         <StorageClass>GLACIER</StorageClass>
         </Transition>
         <NoncurrentVersionTransition>
          <NoncurrentDays>30</NoncurrentDays>
          StorageClass>STANDARD_IA/StorageClass>
         </NoncurrentVersionTransition>
         <NoncurrentVersionTransition>
          <NoncurrentDays>60</NoncurrentDays>
          <StorageClass>GLACIER</StorageClass>
         </NoncurrentVersionTransition>
    </Rule>
</LifecycleConfiguration>
```

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

在此请求返回的响应消息体中包含的配置元素如下表8-16描述。

#### 表 8-16 生命周期配置元素

名称	描述
Date	表示规则生效的时间。 该值必须兼容ISO8601格式,而且必须是UTC午夜0点。 类型:字符串 父节点:Expiration, Transition
Days	表示在对象创建时间后第几天时规则生效。 类型:正整数 父节点: Expiration, Transition
StorageClass	表示对象将被修改成的存储级别。 类型:字符串STANDARD_IA或GLACIER 父节点:Transition,NoncurrentVersionTransition
Transition	生命周期配置中表示迁移时间和迁移后对象存储级别的元素(仅针对对象的最新版本) 类型: XML 子节点: Date或Days 父节点: Rule
Expiration	生命周期配置中表示过期时间的Container。 类型: XML 子节点: Date或Days 父节点: Rule
ID	一条Rule的标识,由不超过255个字符的字符串组成。 类型:字符串 父节点:Rule
LifecycleConfiguration	生命周期配置Rule的Container。可以配置多条Rule,但需保证整个配置消息体总大小不超过20KB。 类型:XML 子节点:Rule 父节点:无
NoncurrentDays	表示对象在成为历史版本之后第几天时规则生效。 类型:正整数 父节点: NoncurrentVersionExpiration, NoncurrentVersionTransition

名称	描述
NoncurrentVersionTransition	生命周期配置中表示对象的历史版本迁移时间和迁移后对象存储级别的元素。 类型: XML 子节点: NoncurrentDays, StorageClass 父节点: Rule
NoncurrentVersionExpiration	生命周期配置中表示历史版本过期时间的Container。您可以将该动作设置在已启用多版本(或暂停)的桶,来让系统删除对象的满足特定生命周期的历史版本。 类型:XML 子节点:NoncurrentDays 父节点:Rule
Prefix	对象名前缀,用以标识哪些对象可以匹配到当前这条Rule。 类型:字符串 父节点:Rule
Rule	具体某一条生命周期配置的Container。 类型:容器 父节点: LifecycleConfiguration
Status	标识当前这条Rule是否启用。 类型:字符串 父节点:Rule 有效值:Enabled,Disabled

# 错误响应消息

此请求可能的特殊错误如下表8-17描述。

#### 表 8-17 特殊错误

错误码	描述	HTTP状态码
NoSuchLifecycleConfigurat ion	桶的生命周期配置不存在	404 Not Found

其余错误已经包含在表4-1中。

# 8.15 获取桶的多版本状态

桶的所有者可以获取指定桶的多版本状态。

如果从未设置桶的多版本状态,则此操作不会返回桶的多版本状态。

#### 请求消息样式

GET /?versioning HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

#### 请求消息参数

该请求消息中不使用消息参数。

#### 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

#### 请求消息元素

该请求消息中不使用消息元素。

#### 响应消息样式

HTTP/1.1 status\_code

Date: date Content-Type: type Content-Length: length

 $$$ \end{cases} $$ \end{cases} $$$ 

## 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应中以消息元素的形式返回桶的多版本状态,元素的具体意义如**表8-18**所示。

#### 表 8-18 响应消息元素

名字	描述	是否必选
VersioningConfiguration	多版本状态信息的元素。 类型:元素	是
Status	标识桶的多版本状态。 类型:枚举值 有效值:Enabled,Suspended	否

#### 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.16 获取桶的消息通知配置

获取指定桶的消息通知配置信息。

为了能成功执行此配置操作,需要确保执行者拥有s3:PutBucketNotification权限。默认情况下只有桶的所有者拥有该权限,但可以通过设置桶策略或用户策略授权给其他用户

#### 请求消息样式

```
GET /?notification HTTP/1.1
Host: bucketname.obs.cn-north-1.myhwclouds.com
Date: date
Authorization: authorization
```

#### 请求消息参数

该请求消息中不使用消息参数。

#### 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

## 请求消息元素

该请求消息中不使用消息元素。

# 响应消息样式

```
HTTP/1.1 status_code
Content-Type: type
Date: date
Content-Length: length
<?xml version="1.0" encoding="UTF-8"?>
<NotificationConfiguration xmlns="http://obs.company.com/doc/2015-06-30/">
    <TopicConfiguration>
        <Id>ConfigurationId</Id>
        <Filter>
            <S3Key>
                <FilterRule>
                     <Name>prefix</Name>
                     <Value>prefix-value</Value>
                </FilterRule>
                <FilterRule>
                     <Name>suffix</Name>
                    <Value>suffix-value</Value>
                </FilterRule>
           </S3Key>
        </Filter>
        <Topic>TopicARN</Topic>
        <Event>event-type</Event>
        <Event>event-type</Event>
    </TopicConfiguration>
    <\! {\tt CloudFunctionConfiguration} \!>
        <Id>ConfigurationId</Id>
```

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

在此请求返回的响应消息体中包含的配置元素如下表8-19描述。

#### 表 8-19 Notification 配置元素

名称	描述
NotificationConfigurati on	配置桶事件通知的元素,如果此项内容为空,说明通知功能处于关闭状态。
	类型: 元素
	父节点: 无
	子节点:一个或多个TopicConfiguration, CloudFunctionConfiguration或者FunctionGraphConfiguration
TopicConfiguration	配置事件通知主题的元素。
	类型: 元素
	父节点: NotificationConfiguration
	子节点:Id,Filter,Topic,一个或多个Event
Topic	事件通知主题的URN,当OBS检测到桶中发生特定的事件 后,将会发布通知消息至该主题。
	类型:字符串
	父节点: TopicConfiguration
CloudFunctionConfigur	配置函数服务的元素。
ation	类型: 元素
	父节点: NotificationConfiguration
	子节点:Id,Filter,Topic,一个或多个Event

名称	描述
CloudFunction	函数服务中函数的URN,当OBS检测到桶中发生特定的事件后,将会调用该函数。 类型:字符串
	文章: 子初中 父节点: CloudFunctionConfiguration
FunctionGraphConfigur ation	配置函数工作流服务的元素。 类型:元素 父节点:NotificationConfiguration 子节点:Id, Filter, Topic,一个或多个Event
FunctionGraph	函数工作流服务中函数工作流的URN,当OBS检测到桶中 发生特定的事件后,将会调用执行该函数工作流。 类型:字符串 父节点:FunctionGraphConfiguration
Id	每项事件通知配置的唯一标识,若是用户未指定ID,系统将自动分配一个ID。 类型:字符串 父节点:TopicConfiguration,CloudFunctionConfiguration或者FunctionGraphConfiguration
Filter	S3Key的元素,用以保存过滤对象名的一组规则。 类型:元素 父节点:TopicConfiguration,CloudFunctionConfiguration或者FunctionGraphConfiguration 子节点:S3Key
S3Key	S3Key的元素,用以保存过滤对象名的一组规则。 类型:元素 父节点:TopicConfiguration,CloudFunctionConfiguration或者FunctionGraphConfiguration
FilterRule	定义过滤规则键值对的元素。 类型:元素 父节点:S3Key 子节点:Name,Value
Name	指定规则按对象名前缀或后缀进行过滤。 类型:字符串 父节点:FilterRule 合法值:prefix或者suffix
Value	指定的对象名关键字,用以按照前缀或后缀过滤对象。 类型:字符串 父节点:FilterRule

名称	描述		
Event	需要发布通知消息的事件类型。		
	<b>说明</b> 在一个TopicConfiguration,CloudFunctionConfiguration或者 FunctionGraphConfiguration配置项中可以添加多个事件类型。		
	类型:字符串		
	合法值:		
	s3:ObjectCreated:*		
	s3:ObjectCreated:Put		
	s3:ObjectCreated:Post		
	s3:ObjectCreated:Copy		
	s3:ObjectCreated:CompleteMultipartUpload		
	s3:ObjectRemoved:*		
	s3:ObjectRemoved:Delete		
	s3:ObjectRemoved:DeleteMarkerCreated		
	父节点: TopicConfiguration, CloudFunctionConfiguration或者FunctionGraphConfiguration		

#### 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.17 获取桶默认存储类型

获取该桶设置的默认存储类型信息。

要正确执行此操作,需要确保执行者有s3:GetBucketStoragePolicy执行权限。默认情况下只有桶的所有者可以执行此操作,也可以通过设置桶策略或用户策略授权给其他用户。

#### 请求消息样式

GET /?storagePolicy HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

该请示消息中不带消息参数。

# 请求消息头

该请求使用公共的请求消息头,具体参见公共请求消息头章节。

# 请求消息元素

该请求消息不使用消息元素。

HTTP/1.1 status\_code

Date: date Content-Type: type Content-Length: length

#### 响应消息头

该请求的响应消息使用公共消息头,具体请参考3公共消息头章节。

## 响应消息元素

该响应以消息元素的形式返回桶的存储级别信息,元素的具体意义如表8-20所示。

#### 表 8-20 响应消息元素

元素名称	描述
StoragePolicy	桶的存储规格,包括桶的默认存储级别。 类型:XML。
DefaultStorageClass	桶的默认存储类型。 类型:字符串,具体枚举类型参考 <b>表8-10</b> 。

#### 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.18 获取桶的跨区域复制配置

获取指定桶的复制配置信息。执行该配置操作前需要确保执行者拥有s3: GetReplicationConfiguration权限。

#### 请求消息样式

GET /?replication HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization:authorization string(see Authenticating Requests (AWS Signature Version4))

# 请求参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

```
HTTP/1.1 status_code
Date: date
Content-Length: contentlength
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://obs.company.com/doc/2006-03-01/">
  <Agency>exampleAgency</Agency>
  <Rule>
      <Status>Enabled</Status>
      <Prefix></Prefix>
      <Destination>
        \label{lem:condition} $$\langle StorageClass\rangle STANDARD_IA \langle /StorageClass\rangle $$
     </Destination>
  </Rule>
</ReplicationConfiguration>
```

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

在此请求返回的响应消息体中包含的配置元素如表8-21所示。

#### 表 8-21 桶的复制配置元素

名称	描述	是否必须
ReplicationConfigu ration	复制规则的容器,最多可以达到100条规则,所有的复制配置大小可达到50KB。	是
	类型: 容器	
	子节点:Rule	
	父节点:无	
Agency	用户创建的委托名字,最大长度为 <b>64</b> 字符。	是
	类型:字符串	
	父节点: ReplicationConfiguration	
Rule	一条特定复制规则信息的容器。	是
	复制配置必须至少配置一条规则,最多 能达到100条规则。	
	类型:容器	
	父节点:	
	ReplicationConfiguration	
ID	规则的特殊辨识符,最大长度为255字符。	否
	类型:字符串	
	父节点:Rule	

名称	描述	是否必须
Status	如果Status为Disabled,这条规则会被忽略。	是
	类型:字符串	
	父节点:Rule	
	有效值: Enabled,Disabled	
Prefix	对象名的前缀,适配于一个或者多个对象。	是
	最大前缀长度可达到为1024个字节,一个前缀不能包含已经存在的另一个前 级。	
	类型:字符串	
	父节点:Rule	
Destination	目标桶信息的容器。	是
	类型:容器	
	父节点:Rule	
Bucket	存储被规则标识的对象副本的桶名称。	是
	如果在复制配置中有多条规则,这些规 则必须都要标识同一个桶作为目标桶。	
	类型:字符串	
	父节点:Destination	
StorageClass	对象的存储类型。	否
	类型: 枚举值	
	父节点:Destination	
	有效值: STANDARD STANDARD_IA  GLACIER	

# 错误响应消息

在此请求的响应中错误响应消息如表8-22所示。

#### 表 8-22 桶的错误响应元素

错误码	描述	HTTP 响应码	SOAP错误码前缀
NoSuchReplicationCon figuration	跨region复制配 置不存在	404 not found	Client

# 8.19 删除桶策略

该接口的实现是通过使用policy子资源来删除一个指定桶上的策略。

要使用该接口,使用者必须是桶的所有者或者是被桶策略授权s3:DeleteBucketPolicy权限的用户。

无论桶的策略本身是否存在,系统都直接返回204"No Content"的结果。

#### 请求消息样式

DELETE /?policy HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

#### 请求消息参数

该请求消息中不使用消息参数。

#### 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

#### 请求消息元素

该请求消息中不使用消息元素。

#### 响应消息样式

HTTP/1.1 status\_code

Date: date

Content-Type: text/xml Content-Length: *length* 

#### 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 响应消息元素

该请求的响应消息中不带有响应元素。

#### 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.20 删除桶的生命周期配置

删除指定桶的生命周期配置信息。删除后桶中的对象不会过期,OBS不会自动删除桶中对象。

要正确执行此操作,需要确保执行者有s3:PutLifecycleConfiguration权限。默认情况下只有桶的所有者可以执行此操作,也可以通过设置桶策略或用户策略授权给其他用户。

#### 请求消息样式

DELETE /?lifecycle HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: Authorization

#### 请求消息参数

该请求消息中不使用消息参数。

## 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

#### 请求消息元素

该请求消息中不使用消息元素。

#### 响应消息样式

HTTP/1.1 status\_code

Date: date

Content-Type: text/xml

Date: date

#### 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 响应消息元素

该请求的响应消息中不带消息元素。

#### 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 8.21 删除桶的跨区域复制配置

删除桶的复制配置。执行该配置操作前需要确保执行者拥有s3: DeleteReplicationConfiguration权限。

#### 请求消息样式

DELETE /?replication HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization string (see Authenticating Requests (AWS Signature Version4))

# 请求参数

该请求消息中不使用消息参数。

## 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

HTTP/1.1 204 No Content

Server: OBS Date: date

Connection: keep-alive

# 响应消息头

该请求的响应消息使用公共消息头,具体请参见公共请求消息头章节。

# 响应消息元素

该请求的响应消息中不带有响应元素。

#### 错误响应消息

在此请求的响应中不会返回特殊错误。

# 9 静态网站托管

- 9.1 设置桶的网站配置
- 9.2 设置桶的CORS配置
- 9.3 OPTIONS桶
- 9.4 OPTIONS对象
- 9.5 获取桶的网站配置
- 9.6 获取桶的CORS配置
- 9.7 删除桶的网站配置
- 9.8 删除桶的CORS配置

# 9.1 设置桶的网站配置

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。

本接口实现为桶创建或更新网站配置信息。

要正确执行此操作,需要确保执行者有s3:PutBucketWebsite权限。默认情况下只有桶的所有者可以执行此操作,也可以通过设置桶策略或用户策略授权给其他用户。

#### □ 说明

尽量避免目标桶名中带有".",否则通过HTTPS访问时可能出现客户端校验证书出错。

#### 请求消息样式

PUT /?website HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Content-Length: *length* 

Date: date

Authorization: authorization

#### 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

# 请求消息元素

在此请求中,需要在请求的消息体中配置桶的网站配置信息,配置信息以XML格式上传。

● 如果重定向所有请求,网站配置元素如表9-1描述。

#### 表 9-1 重定向所有请求 Website 配置元素

名称	描述	是否必选
WebsiteConfiguration	网站配置的根节点。 类型:元素 父节点:无	是
RedirectAllRequestsTo	描述所有请求的重定向行为,如果这个节点出现,所有其他的兄弟节点都不能出现。 类型:元素 父节点:WebsiteConfiguration	是
HostName	描述重定向的站点名。 类型:字符串 父节点:RedirectAllRequestsTo	是
Protocol	描述重定向请求时使用的协议(http,https),默认使用http协议。 类型:字符串 父节点:RedirectAllRequestsTo	否

● 如果想要设定重定向规则,网站配置元素如**表9-2**描述。

#### 表 9-2 设定重定向规则 Website 配置元素

名称	描述	是否必选
WebsiteConfiguration	Website配置的根节点。 类型:元素 父节点:无	是
IndexDocument	Suff元素。 类型:元素 父节点:WebsiteConfiguration	是
Suffix	Suffix 元素被追加在对文件夹的请求的末尾(例如: Suffix 配置的是 "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)。Suffix 元素不能为空或者包含 "/"字符。 类型:字符串 父节点: IndexDocument	是
ErrorDocument	Key 元素。类型: 元素父节点: WebsiteConfiguration	否
Key	当4XX错误出现时使用的对象的名称。这个元素指定了当错误出现时返回的页面。 类型:字符串 父节点:ErrorDocument 条件:父节点ErrorDocument存在时	否
RoutingRules	Routin元素。 类型:元素 父节点:WebsiteConfiguration	否
RoutingRule	重定向规则的元素。一条重定向规则包含一个Condition和一个Redirect,当Condition匹配时,Redirect生效。 类型:元素 父节点:RoutingRules 元素中至少要有一个RoutingRule元素	是
Condition	描述重定向规则匹配的条件的元素。 类型:元素 父节点:RoutingRule	否

名称	描述	是否必选
KeyPrefixEquals	描述当重定向生效时对象名的前缀。例如:  ● 重定向ExamplePage.html对象的请求,	否
HttpErrorCodeReturnedEquals	描述Redirect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Redirect生效。例如:  • 当返回的http错误码为404时重定向到NotFound.html,可以将Condition中的HttpErrorCodeReturnedEquals设置为404,Redirect中的ReplaceKeyWith设置为NotFound.html。 类型:字符串 父节点:Condition 条件:父节点Condition存在,并且兄弟节点KeyPrefixEquals不存在。如果设定了多个条件,需要同时匹配所有的条件,Redirect才可生效。	否
Redirect	重定向信息的元素。可以重定向到另一个站点、另一个页面或使用另一个协议。当事件或错误发生时,可以指定不同的返回码。 类型:元素 父节点:RoutingRule	是
Protocol	描述重定向请求时使用的协议。 类型:字符串 父节点:Redirect 可选值:http、https 条件:有其他兄弟节点存在时非必选	否
HostName	描述重定向请求时使用的站点名。 类型:字符串 父节点:Redirect 条件:有其他兄弟节点存在时非必选	否

名称	描述	是否必选
ReplaceKeyPrefixWith	描述重定向请求时使用的对象名前缀。例如:  • 想把所有对docs(目录下的对象)的请求重定向到documents(目录下的对象),可以将Condition中的KeyPrefixEquals设置为docs,Redirect中的ReplaceKeyPrefixWith设置为documents。 类型:字符串 父节点:Redirect 条件:有其他兄弟节点存在时非必选,不可	否
	与ReplaceKeyWith同时存在	
ReplaceKeyWith	描述重定向请求时使用的对象名。例如:重定向请求到error.html。 类型:字符串 父节点:Redirect 条件:有其他兄弟节点存在时非必选,不可与ReplaceKeyPrefixWith同时存在	否
HttpRedirectCode	描述响应中的HTTP状态码。 类型:字符串 父节点:Redirect 条件:有其他兄弟节点存在时非必选	否

HTTP/1.1 status\_code

Date: date

Content-Length: length

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应消息不带消息元素。

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 9.2 设置桶的 CORS 配置

CORS(Cross Origin Resource Sharing),即跨域资源共享,是W3C标准化组织提出的一种规范机制,允许客户端的跨域请求的配置。在通常的网页请求中,由于安全策略 SOP(Same Origin Policy)的存在,一个网站的脚本和内容是不能与另一个网站的脚本和内容发生交互的。

OBS允许在桶内保存静态的网页资源,在正确的使用下,OBS的桶可以成为网站资源(请参见9.1 设置桶的网站配置)。只有进行了适当的CORS配置,OBS中的网站才能响应另一个网站的跨域请求。

典型的应用场景如下:

- 你可以使用CORS支持,使用JavaScript和HTML 5来构建Web应用,直接访问OBS中的资源,而不再需要代理服务器做中转。
- 可以使用HTML 5中的拖拽功能,直接向OBS上传文件,展示上传进度,或是直接 从Web应用中更新内容。
- 托管在不同域中的外部网页、样式表和HTML 5应用,现在可以引用存储在OBS中的Web字体或图片,让这些资源能被多个网站共享。

要正确执行此操作,需要确保执行者有s3:PutBucketCORS权限。默认情况下只有桶的 所有者可以执行此操作,也可以通过设置桶策略或用户策略授权给其他用户。

#### 请求消息样式

```
PUT /?cors HTTP/1.1
Host: bucketname.obs.cn-north-1.myhwclouds.com
Content-Length: length
Date: date
Authorization: authorization
Content-MD5: MD5
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration>
    <CORSRule>
        <TD>id</TD>
        <AllowedMethod>method</AllowedMethod>
        <AllowedOrigin>origin</AllowedOrigin>
        <AllowedHeader>header</AllowedHeader>
        <MaxAgeSeconds>seconds</MaxAgeSeconds>
        <ExposeHeader>header/ExposeHeader>
    </CORSRule>
</CORSConfiguration>
```

# 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用的消息头如下表9-3所示。

#### 表 9-3 CORS 请求消息头

消息头名称	描述	是否必选
Content-MD5	按照RFC 1864标准计算出消息体的MD5摘要字符串,即消息体128-bit MD5值经过base64编码后得到的字符串。	是
	类型:字符串	
	示例: n58IG6hfM7vqI4K0vnWpog==	

# 请求消息元素

在此请求中,需要在请求的消息体中配置桶的CORS配置信息。配置信息以XML格式上传,具体的配置元素如表9-4描述。

#### 表 9-4 CORS 配置元素

名称	描述	是否必选
CORSConfigur ation	CORSRules的根节点,最大不超过64 KB。。 类型:容器。 父节点:无。	是
CORSRule	CORS规则,CORSConfiguration下可最多包含100个规则。 类型:容器。 父节点: CORSConfiguration。	是
ID	一条Rule的标识,由不超过255个字符的字符串组成。 类型:字符串。 父节点:Rule。	否
AllowedMetho d	CORS规则允许的Method。 类型:字符串。 有效值:GET、PUT、HEAD、POST、DELETE 父节点:Rule。	是
AllowedOrigin	CORS规则允许的Origin(表示域名的字符串),可以带一个匹配符"*"。每一个AllowedOrigin可以带最多一个"*"通配符。 类型:字符串。 父节点: Rule。	是

名称	描述	是否必选
AllowedHeader	配置CORS请求中允许携带的"Access-Control-Request-Headers"头域。如果一个请求带了 "Access-Control-Request-Headers"头域,则只有匹配上AllowedHeader中的配置才认为是一个合法的CORS请求。每一个AllowedHeader可以带最多一个"*"通配符,不可出现空格。 类型:字符串。 父节点:Rule。	否
MaxAgeSecond s	客户端可以缓存的CORS响应时间,以秒为单位。 每个CORSRule可以包含至多一个MaxAgeSeconds, 可以设置为负值。 类型:整数。 父节点: Rule。	否
ExposeHeader	CORS响应中带的附加头域,给客户端提供额外的信息,不可出现空格。 类型:字符串。 父节点:Rule。	否

HTTP/1.1 status\_code

Date: date

Content-Length: length

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

## 响应消息元素

该请求的响应消息不带消息元素。

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 9.3 OPTIONS 桶

OPTIONS,称为预请求,是客户端发送给服务端的一种请求,通常被用于检测客户端是否具有对服务端进行操作的权限。只有当预请求成功返回,客户端才开始执行后续的请求。

OBS允许在桶内保存静态的网页资源,在正确的使用下,OBS的桶可以成为网站资源。在这种使用场景下,OBS中的桶作为服务端,需要处理客户端发送的OPTIONS预请求。

要处理OPTIONS,OBS的桶必须已经配置CORS,关于CORS的使用说明,请参见章节 9.2 设置桶的CORS配置。

#### 请求消息样式

OPTIONS / HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

Origin: origin

Access-Control-Request-Method: method

## 请求消息参数

该请求消息中不使用消息参数。

#### 请求消息头

该请求使用的消息头如下表9-5所示。

#### 表 9-5 OPTIONS 请求消息头

消息头名称	描述	是否必选
Origin	预请求指定的跨域请求Origin(通常为域名)。 类型:字符串	是
Access- Control- Request- Method	实际请求可以带的HTTP方法,可以带多个方法头域。 类型:字符串 有效值:GET、PUT、HEAD、POST、DELETE	是
Access- Control- Request- Headers	实际请求可以带的HTTP头域,可以带多个头域。 类型:字符串	否

# 请求消息元素

该请求消息中不使用消息元素。

#### 响应消息样式

HTTP/1.1 status\_code

Content-Type: application/xml
Access-Control-Allow-Origin: origin
Access-Control-Allow-Methods: method
Access-Control-Allow-Header: header
Access-Control-Max-Age: time

Access-Control-Expose-Headers: header

Date: date

Content-Length: length

#### 响应消息头

该响应使用的消息头如下表9-6所示。

#### 表 9-6 CORS 响应消息头

消息头名称	描述
Access-Control-Allow-	如果请求的Origin满足服务端的CORS配置,则在响应中包含这个Origin。
Origin	类型:字符串
Access-Control-Allow-	如果请求的headers满足服务端的CORS配置,则在响应中包含这个headers。
Headers	类型:字符串
Access-Control-Max-	服务端CORS配置中的MaxAgeSeconds。
Age	类型:整数
Access-Control-Allow-Methods	如果请求的Access-Control-Request-Method满足服务端的CORS配置,则在响应中包含这条rule中的Methods。 类型:字符串 有效值:GET、PUT、HEAD、POST、DELETE
Access-Control-	服务端CORS配置中的ExposeHeader。
Expose-Headers	类型:字符串

## 响应消息元素

该请求的响应消息中不带消息元素。

# 错误响应消息

此请求可能的特殊错误如下表9-7描述。

#### 表 9-7 特殊错误

错误码	描述	HTTP状态码
Bad Request	Invalid Access-Control-Request-Method: null 桶配置了cors,OPTIONS桶时,没有加入method头域。	400 BadRequest
Bad Request	Insufficient information. Origin request header needed. 桶配置了cors,OPTIONS桶时,没有加入origin头域。	400 BadRequest

错误码	描述	HTTP状态码
AccessForbidden	CORSResponse: This CORS request is not allowed. This is usually because the evalution of Origin, request method / Access-Control-Request-Method or Access-Control-Requet-Headers are not whitelisted by the resource's CORS spec. 桶配置了cors,OPTIONS桶时,Origin、method、Headers与任一rule匹配不上。	403 Forbidden

其余错误已经包含在表4-1中。

# 9.4 OPTIONS 对象

请参见章节 9.3 OPTIONS桶。

#### 请求消息样式

OPTIONS /object HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

Origin: origin

Access-Control-Request-Method: method

#### 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用的消息头如下表9-8所示。

#### 表 9-8 OPTIONS 请求消息头

消息头名称	描述	是否必选
Origin	预请求指定的跨域请求Origin(通常为域名)。 类型:字符串	是
Access- Control- Request- Method	实际请求可以带的HTTP方法,可以带多个方法头域。 类型:字符串 有效值:GET、PUT、HEAD、POST、DELETE	是
Access- Control- Request- Headers	实际请求可以带的HTTP头域,可以带多个头域。 类型:字符串	否

#### 请求消息元素

该请求消息中不使用消息元素。

#### 响应消息样式

HTTP/1.1 status\_code

Content-Type: type

Access-Control-Allow-Origin: origin Access-Control-Allow-Methods: method Access-Control-Allow-Header: header Access-Control-Max-Age: time

Access-Control-Expose-Headers: header

Date: date

Content-Length: length

#### 响应消息头

该请求使用的消息头如下表9-9所示。

#### 表 9-9 CORS 请求消息头

消息头名称	描述
Access-Control-Allow-	如果请求的Origin满足服务端的CORS配置,则在响应中包含这个Origin。
Origin	类型:字符串
Access-Control-Allow-	如果请求的headers满足服务端的CORS配置,则在响应中包含这个headers。
Headers	类型:字符串
Access-Control-Max-	服务端CORS配置中的MaxAgeSeconds。
Age	类型:整数
Access-Control-Allow-Methods	如果请求的Access-Control-Request-Method满足服务端的CORS配置,则在响应中包含这条rule中的Methods。 类型:字符串 有效值:GET、PUT、HEAD、POST、DELETE
Access-Control-	服务端CORS配置中的ExposeHeader。
Expose-Headers	类型:字符串

# 响应消息元素

该请求的响应消息中不带消息元素。

#### 错误响应消息

此请求可能的特殊错误如下表9-10描述。

#### 表 9-10 特殊错误

错误码	描述	HTTP状态码
Bad Request	Invalid Access-Control-Request-Method: null 桶配置了cors,OPTIONS桶时,没有加入method头域。	400 BadRequest
Bad Request	Insufficient information. Origin request header needed. 桶配置了cors,OPTIONS桶时,没有加入origin头域。	400 BadRequest
AccessForbidden	CORSResponse: This CORS request is not allowed. This is usually because the evalution of Origin, request method / Access-Control-Request-Method or Access-Control-Requet-Headers are not whitelisted by the resource's CORS spec. 桶配置了cors,OPTIONS桶时,Origin、method、Headers与任一rule匹配不上。	403 Forbidden

其余错误已经包含在表4-1中。

# 9.5 获取桶的网站配置

获取该桶设置的网站配置信息。

要正确执行此操作,需要确保执行者有s3:GetBucketWebsite执行权限。默认情况下只有桶的所有者可以执行此操作,也可以通过设置桶策略或用户策略授权给其他用户。

## 请求消息样式

GET /?website HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

## 请求消息元素

该请求消息中不使用消息元素。

#### 响应消息样式

```
HTTP/1.1 status_code

Date: date

Content-Type: type

Content-Length: length

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<WebsiteConfiguration xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">

<RedirectAllRequestsTo>

<HostName>hostName</HostName>

</RedirectAllRequestsTo>

</WebsiteConfiguration>
```

#### 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 响应消息元素

在此请求返回的响应消息体中包含的配置元素与设置桶的网站配置请求的请求消息元素一致,见**请求消息元素**。

#### 错误响应消息

此请求可能的特殊错误如下表9-11描述。

#### 表 9-11 特殊错误

错误码	描述	HTTP状态码
NoSuchWebsiteConfiguration	桶的Website配置不存在	404 Not Found

其余错误已经包含在表4-1中。

# 9.6 获取桶的 CORS 配置

获取指定桶的CORS配置信息。

要正确执行此操作,需要确保执行者有s3:GetBucketCORS权限。默认情况下只有桶的 所有者可以执行此操作,也可以通过设置桶策略或用户策略授权给其他用户。

#### 请求消息样式

GET /?cors HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

该请求消息中不使用消息参数。

## 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

#### 请求消息元素

该请求消息中不使用消息元素。

#### 响应消息样式

```
HTTP/1.1 status_code

Content-Type: application/xml

Date: date

Content-Length: length
```

## 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 响应消息元素

在此请求返回的响应消息体中包含的配置元素如下表9-12描述。

#### 表 9-12 CORS 配置元素

名称	描述
CORSConfiguration	CORSRules的根节点,最大不超过64 KB。。 类型:容器。 父节点:无。
CORSRule	CORS规则,CORSConfiguration下可最多包含100个规则。 类型:容器。 父节点:CORSConfiguration。
ID	一条Rule的标识,由不超过255个字符的字符串组成。 类型:字符串。 父节点:Rule。
AllowedMethod	CORS规则允许的Method。 类型:字符串。 有效值:GET、PUT、HEAD、POST、DELETE 父节点:Rule。
AllowedOrigin	CORS规则允许的Origin(表示域名的字符串),可以带一个匹配符"*"。每一个AllowedOrigin可以带最多一个"*"通配符。 类型:字符串。 父节点:Rule。

名称	描述
AllowedHeader	配置CORS请求中允许携带的"Access-Control-Request-Headers"头域。如果一个请求带了"Access-Control-Request-Headers"头域,则只有匹配上AllowedHeader中的配置才认为是一个合法的CORS请求。每一个AllowedHeader可以带最多一个"*"通配符,不可出现空格。 类型:字符串。
MaxAgeSeconds	客户端可以缓存的CORS响应时间,以秒为单位。 每个CORSRule可以包含至多一个MaxAgeSeconds,可以设置为负值。 类型:整数。 父节点: Rule。
ExposeHeader	CORS响应中带的附加头域,给客户端提供额外的信息,不可出现空格。 类型:字符串。 父节点:Rule。

#### 错误响应消息

此请求可能的特殊错误如下表9-13描述。

#### 表 9-13 特殊错误

错误码	描述	HTTP状态码
NoSuchCORSConfiguration	桶的CORS配置不存在	404 Not Found

其余错误已经包含在表4-1中。

# 9.7 删除桶的网站配置

删除指定桶的网站配置信息。

要正确执行此操作,需要确保执行者有s3:DeleteBucketWebsite权限。默认情况下只有桶的所有者可以执行此操作,也可以通过设置桶策略或用户策略授权给其他用户。

#### 请求消息样式

DELETE /?website HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

#### 请求消息参数

该请求消息中不使用消息参数。

## 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

#### 请求消息元素

该请求消息中不使用消息元素。

#### 响应消息样式

HTTP/1.1 status\_code

Date: date

Content-Type: type Content-Length: length

#### 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 响应消息元素

该请求的响应消息中不带消息元素。

#### 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 9.8 删除桶的 CORS 配置

删除指定桶的CORS配置信息。删除后桶以及桶中的对象将不能再被其他网址发送的请求访问。

要正确执行此操作,需要确保执行者有s3:PutBucketCORS权限。默认情况下只有桶的 所有者可以执行此操作,也可以通过设置桶策略或用户策略授权给其他用户。

#### 请求消息样式

DELETE /?cors HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该请求使用公共消息头,具体参见公共请求消息头章节。

#### 请求消息元素

该请求消息中不使用消息元素。

## 响应消息样式

HTTP/1.1 status\_code

Date: date

Content-Type: application/xml Content-Length: *length* 

## 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 响应消息元素

该请求的响应消息中不带消息元素。

## 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# **10** 对象操作

- 10.1 PUT上传
- 10.2 POST上传
- 10.3 复制对象
- 10.4 设置对象ACL
- 10.5 获取对象内容
- 10.6 获取对象元数据
- 10.7 获取对象ACL
- 10.8 删除对象
- 10.9 批量删除对象
- 10.10 取回归档存储对象

# 10.1 PUT 上传

上传对象操作是指在指定的桶内增加一个对象,执行该操作需要用户拥有桶的写权限。

#### ∭说明

用户上传的对象存储在桶中。用户必须对桶有WRITE权限,才可以在桶中上传对象。同一个桶中存储的对象名必须是唯一的。

如果在指定的桶内已经有相同的对象键值的对象,用户上传的新对象会覆盖原来的对象;为了确保数据在传输过程中没有遭到破坏,用户可以在请求消息头中加入Content-MD5参数。在这种情况下,OBS收到上传的对象后,会对对象进行MD5校验,如果不一致则返回出错信息。用户还可以在上传对象时指定x-amz-acl参数,设置对象的权限控制策略。

该操作支持服务端加密功能。

用户在OBS系统中创建了桶之后,可以采用PUT操作的方式将对象上传到桶中。

#### 多版本

如果桶的多版本状态是开启的,系统会自动为对象生成一个唯一的版本号,并且会在响应报头x-amz-version-id返回该版本号。如果桶的多版本状态是暂停的,则对象的版本号为null。关于桶的多版本状态,参见8.6 设置桶的多版本状态。

## 请求消息样式

PUT /ObjectName HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Content-Type: application/xml Content-Length: *length* Authorization: *authorization* Date: *date* 

<Optional Additional Header>

<object Content>

#### 请求消息参数

该请求消息中不使用参数。

## 请求消息头

该请求可以使用附加的消息头,具体如表10-1所示。

#### 表 10-1 请求消息头

消息头名称	描述	是否 必选
Content-MD5	按照RFC 1864标准计算出消息体的MD5摘要字符串,即消息体128-bit MD5值经过base64编码后得到的字符串。 类型:字符串	否
	示例: n58IG6hfM7vqI4K0vnWpog==。	
x-amz-acl	创建对象时,可以加上此消息头设置对象的权限控制策略,使用的策略为预定义的常用策略,包括: private; public-read; public-read-write; authenticated-read; bucket-owner-read; bucket-owner-full-control(各策略详细说明见表6-3)。	否
	说明:字符串形式的预定义策略。	
x-amz-storage-class	创建对象时,可以加上此头域设置对象的存储类型。 如果未设置此头域,则以桶的默认存储类型作为对象 的存储类型。 类型:字符串	否
	说明:存储类型有3种:STANDARD(标准存储)、STANDARD_IA(低频访问存储)、GLACIER(归档存储)。因此这里可配置的值有:STANDARD、STANDARD_IA和GLACIER,注意大小写敏感。	
	示例: x-amz-storage-class: STANDARD	

消息头名称	描述	是否 必选
x-amz-meta-*	创建对象时,可以在HTTP请求中加入以"x-amz-meta-"开头的消息头,用来加入自定义的元数据,以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时,加入的自定义元数据将会在返回消息的头中出现。HTTP请求不包含消息体,长度不能超过8KB。 类型:字符串示例:x-amz-meta-test:test metadata	否
x-amz-website-redirect-location	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,OBS将 这个值从头域中取出,保存在对象的元数据中。 例如,重定向请求到桶内另一对象: x-amz-website-redirect-location:/anotherPage.html 或重定向请求到一个外部URL: x-amz-website-redirect-location:http://www.huawei.com/ 类型:字符串 默认值:无 约束:必须以"/"、"http://"或"https://"开头,长 度不超过2K。	否
x-amz-server-side- encryption	使用该头域表示服务端加密是SSE-KMS方式。 类型:字符串 示例: x-amz-server-side-encryption: aws:kms	否。 当用 SSE- KMS 方时 必 选。
x-amz-server-side- encryption-aws-kms- key-id	SSE-KMS方式下使用该头域,该头域表示主密钥,如果用户没有提供该头域,那么默认的主密钥将会被使用。 类型:字符串 示例:x-amz-server-side-encryption-aws-kms-key-id:arn:aws:kms:sichuan:domainiddomainiddomainiddoma0001:key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0	否

消息头名称	描述	是否 必选
x-amz-server-side- encryption- customer-algorithm	SSE-C方式下使用该头域,该头域表示加密使用的算法。 类型:字符串 示例: x-amz-server-side-encryption-customer- algorithm: AES256 约束: 需要和x-amz-server-side-encryption-customer- key, x-amz-server-side-encryption-customer- key. x-dmz-server-side-encryption-customer-key-MD5 一起使用。	否当用 SSE- C式 时必
x-amz-server-side- encryption- customer-key	SSE-C方式下使用该头域,该头域表示加密使用的密钥。该密钥用于加密对象。 类型:字符串示例:x-amz-server-side-encryption-customer-key:K7QkYpBkM5+hcs27fsNkUnNVaobncnLht/rCB2o/9Cw=约束:该头域由256-bit的密钥经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-key-MD5一起使用。	否当用 SSE- C式时必选。
x-amz-server-side- encryption- customer-key-MD5	SSE-C方式下使用该头域,该头域表示加密使用的密钥的MD5值。MD5值用于验证密钥传输过程中没有出错。 类型:字符串示例:x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==约束:该头域由密钥的128-bit MD5值经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-key一起使用。	否当用 SSE- C式时必选。

# 请求消息元素

该请求消息中不使用消息元素,在消息体中带的是对象的数据。

## 响应消息样式

HTTP/1.1 status\_code Content-Length: length

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

除公共响应消息头之外,还可能使用如表10-2中的消息头。

#### 表 10-2 附加响应消息头

消息头名称	描述
x-amz-version-id	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。 类型:字符串
x-amz-server-side-encryption	如果服务端加密是SSE-KMS方式,响应包含该头域。 类型:字符串 示例: x-amz-server-side-encryption: aws:kms
x-amz-server-side-encryption-aws- kms-key-id	如果服务端加密是SSE-KMS方式,响应包含该头域,该头域表示主密钥。 示例: x-amz-server-side-encryption-aws-kms-key-id: arn:aws:kms:sichuan:domainiddomainiddomainiddoma0001:key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0
x-amz-server-side-encryption- customer-algorithm	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示加密使用的算法。 类型:字符串 示例: x-amz-server-side-encryption-customer- algorithm: AES256
x-amz-server-side-encryption- customer-key-MD5	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示加密使用的密钥的MD5值。 类型:字符串 示例: x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==
x-amz-storage-class	对象为非标准存储对象时,会返回此头域,可取值为: STANDARD_IA或者GLACIER 类型: 字符串

## 响应消息元素

该请求的响应消息不带消息元素。

## 错误响应消息

该请求的返回无特殊错误,所有错误已经包含在表4-1中。

# 10.2 POST 上传

上传对象操作是指在指定的桶内增加一个对象,执行该操作需要用户拥有桶的写权限。

#### □ 说明

用户上传的对象存储在桶中。用户必须对桶有WRITE权限,才可以在桶中上传对象。同一个桶中存储的对象名必须是唯一的。

如果在指定的桶内已经有相同的对象键值的对象,用户上传的新对象会覆盖原来的对象;为了确保数据在传输过程中没有遭到破坏,用户可以在请求消息头中加入Content-MD5参数。在这种情况下,OBS收到上传的对象后,会对对象进行MD5校验,如果不一致则返回出错信息。用户还可以在上传对象时指定x-amz-acl参数,设置对象的权限控制策略。

用户除了可以用PUT直接上传对象外,还可以使用POST上传对象。

该操作支持服务端加密功能。

#### 多版本

如果桶的多版本状态是开启的,系统会自动为对象生成一个唯一的版本号;如果桶的多版本状态是暂停的,则系统生成的对象版本号为**null**,并由响应报头**x**-amz-version-id返回该版本号。关于桶的多版本状态,参见**8.6 设置桶的多版本状态**。

#### 请求消息样式

```
POST / HTTP/1.1
Host: bucketname. obs. cn-north-1. mvhwclouds. com
User-Agent: browser_data
Accept: file_types
Accept-Language: Regions
Accept-Encoding: encoding
Accept-Charset: character_set
Keep-Alive: 300
Connection: keep-alive
Content-Type: multipart/form-data; boundary=-9431149156168
Content-Length: length
--9431149156168
Content-Disposition: form-data; name="key"
--9431149156168
Content-Disposition: form-data; name="success_action_redirect"
success redirect
--9431149156168
Content-Disposition: form-data; name="content-Type"
content_type
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-uuid"
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-tag"
metadata
--9431149156168
Content-Disposition: form-data; name="AWSAccessKeyId"
access-kev-id
--9431149156168
Content-Disposition: form-data; name="policy"
encoded_policy
--9431149156168
Content-Disposition: form-data; name="signature"
signature=
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename"
Content-Type: image/jpeg
file content
--9431149156168
```

Content-Disposition: form-data; name="submit" Upload to OBS --9431149156168--

#### 请求消息参数

该请求消息中不使用参数。

## 请求消息头

该请求使用公共的消息头,具体请参见公共请求消息头章节。

如果想要获取CORS配置信息,则需要使用的消息头如下表10-3所示。

#### 表 10-3 获取 CORS 配置的请求消息头

消息头名称	描述	是否必选
Origin	预请求指定的跨域请求Origin(通常为域名)。 类型:字符串	是
Access-Control- Request-Headers	实际请求可以带的HTTP头域,可以带多个头域。 类型:字符串	否

## 请求消息元素

该请求消息的消息元素以表单形式组织,表单字段的具体含义如表10-4所示。

#### 表 10-4 请求消息表单元素

参数名称	描述	是否 必选
file	上传的对象的内容。 类型:二进制或文本类型。 约束条件:此参数必须为最后一个参数,否则此 参数之后的参数会被丢弃;一个请求中只能含有 一个file参数。	是
key	通过此请求创建的对象名称。 类型:字符串。	是
AWSAccessKeyId	用来指明请求发起者的Access Key。 类型:字符串。 约束条件:如果该请求包括安全策略参数,则必 须包括此参数。	否
policy	该请求的安全策略描述。 类型:字符串。	否

参数名称	描述	是否 必选
expires	表示上传对象的超期时间,单位是毫秒,具体参考RFC 2616。OBS将此参数值保存下来,当用户下载此对象或HEAD Object时,在响应头中携带此参数。 类型:字符串。 示例: POLICY中:{"expires":"1000"} HTML中: <input name="expires" type="text" value="1000"/>	否
x-amz-acl	创建对象时,可以加上此消息头设置对象的权限控制策略,使用的策略为预定义的常用策略,包括: private; public-read; public-read-write; authenticated-Read; bucket-owner-read; bucket-owner-full-control(各策略详细说明见表6-3)。类型:字符串。示例: POLICY中:{"acl":"public-read"}, HTML中: <input name="acl" type="text" value="public-read"/>	否
x-amz-storage-class	创建对象时,可以加上此头域设置对象的存储类型。如果未设置此头域,则以桶的默认存储类型作为对象的存储类型。 类型:字符串 说明:存储类型有3种:STANDARD(标准存储)、STANDARD_IA(低频访问存储)、 GLACIER(归档存储)。因此这里可配置的值有:STANDARD、STANDARD_IA和GLACIER,注意大小写敏感。 示例: POLICY中:{"storage-class":"STANDARD"}, HTML中: <input name="x-amz-storage-class" type="text" value="STANDARD"/>	否
Cache-Control, Content-Type, Content-Disposition, Content-Encoding	这4个参数是HTTP标准消息头,OBS将这些参数记录下来,当用户下载此对象或Head Object时,在响应消息头中携带这些参数。 类型:字符串。 示例: POLICY中:["starts-with","\$Content-Type","text/"], HTML中: <input name="content-type" type="text" value="text/plain"/>	否

参数名称	描述	是否 必选
success_action_redirect	此参数的值是一个URL,用于指定当此次请求操作成功响应后的重定向的地址。      如果此参数值有效且操作成功,响应码为303,Location头域由此参数以及桶名、对象名、对象的ETag组成。      如果此参数值无效,则OBS忽略此参数的作用,响应码为204,Location头域为对象地址。 类型:字符串。 示例: POLICY中:{"success_action_redirect":"http://123458.com"}, HTML中: <input name="success_action_redirect" type="text" value="http://123458.com"/>	否
x-amz-meta-*	创建对象时,可以在HTTP请求中加入"x-amz-meta-"消息头或以"x-amz-meta-"开头的消息头,用来加入自定义的元数据,以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时,加入的自定义元数据将会在返回消息的消息头中出现。类型:字符串。示例: POLICY中:{"x-amz-meta-test":"test metadata"}, HTML中: <input name="x-amz-meta-test" type="text" value="test metadata"/>	否
success_action_status	这个参数指定成功响应的状态码,允许设定的值为200,201,204。  ● 如果此参数值被设定为200或204,OBS响应消息中body为空。  ● 如果此参数值被设定为201,则OBS响应消息中包含一个XML文档描述此次请求的响应。  ● 当请求不协带此参数或参数无效时,OBS响应码为204。 类型:字符串。 示例: POLICY中:["starts-with","\$success_action_status",""], HTML中: <input name="success_action_status" type="text" value="200"/>	否

参数名称	描述	是否 必选
x-amz-website-redirect-location	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,OBS将这个值从头域中取出,保存在对象的元数据中。 默认值:无约束:必须以"/"、"http://"或"https://"开头,长度不超过2K。	否
x-amz-server-side- encryption	使用该头域表示服务端加密是SSE-KMS方式。 类型:字符串 示例:x-amz-server-side-encryption:aws:kms	否。 当用 SSE- KMS 方时 必 选。
x-amz-server-side- encryption-aws-kms- key-id	SSE-KMS方式下使用该头域,该头域表示主密 钥,如果用户没有提供该头域,那么默认的主密 钥将会被使用。 类型:字符串 示例:x-amz-server-side-encryption-aws-kms-key- id: arn:aws:kms:sichuan:domainiddomainiddom a0001:key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0	否
x-amz-server-side- encryption-customer- algorithm	SSE-C方式下使用该头域,该头域表示加密使用的算法。 类型:字符串 示例: x-amz-server-side-encryption-customer- algorithm: AES256 约束: 需要和x-amz-server-side-encryption- customer-key, x-amz-server-side-encryption- customer-key-MD5一起使用。	否。 当用 SSE-C 方时 必 选。
x-amz-server-side- encryption-customer-key	SSE-C方式下使用该头域,该头域表示加密使用的密钥。该密钥用于加密对象。 类型:字符串示例:x-amz-server-side-encryption-customer-key:K7QkYpBkM5+hcs27fsNkUnNVaobncnLht/rCB2o/9Cw=约束:该头域由256-bit的密钥经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-key-MD5一起使用。	否当用 SSE-C 方时必选。

参数名称	描述	是否 必选
x-amz-server-side- encryption-customer- key-MD5	SSE-C方式下使用该头域,该头域表示加密使用的密钥的MD5值。MD5值用于验证密钥传输过程中没有出错。 类型:字符串示例:x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==约束:该头域由密钥的128-bit MD5值经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-key一起使用。	否当用 SSE-C 方时必选。

## 响应消息样式

HTTP/1.1 status\_code
Content-Type: application/xml

Location: location

Date: date ETag: etag

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

除公共响应消息头之外,还可能使用如表10-5中的消息头。

#### 表 10-5 附加响应消息头

消息头名称	描述
x-amz-version-id	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。如果桶的多版本状态为暂 停,则会返回 <b>null</b> 。 类型:字符串
Access-Control-Allow-Origin	当桶设置了CORS配置,如果请求的Origin满足服务端的CORS配置,则在响应中包含这个Origin。 类型:字符串
Access-Control-Allow-Headers	当桶设置了CORS配置,如果请求的headers满足服务端的CORS配置,则在响应中包含这个headers。 类型:字符串
Access-Control-Max-Age	当桶设置了CORS配置,服务端CORS配置中的 MaxAgeSeconds。 类型:整数

消息头名称	描述
Access-Control-Allow-Methods	当桶设置了CORS配置,如果请求的Access-Control-Request-Method满足服务端的CORS配置,则在响应中包含这条rule中的Methods。 类型:字符串 有效值:GET、PUT、HEAD、POST、DELETE
Access-Control-Expose-Headers	当桶设置了CORS配置,服务端CORS配置中的ExposeHeader。 类型:字符串
x-amz-server-side-encryption	如果服务端加密是SSE-KMS方式,响应包含该头域。 类型:字符串 示例: x-amz-server-side-encryption: aws:kms
x-amz-server-side-encryption-aws- kms-key-id	如果服务端加密是SSE-KMS方式,响应包含该头域,该头域表示主密钥。 示例: x-amz-server-side-encryption-aws-kms-key-id: arn:aws:kms:sichuan:domainiddomainiddomainiddoma0001:key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0
x-amz-server-side-encryption- customer-algorithm	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示加密使用的算法。 类型:字符串 示例: x-amz-server-side-encryption-customer- algorithm: AES256
x-amz-server-side-encryption- customer-key-MD5	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示加密使用的密钥的MD5值。 类型:字符串 示例:x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==

## 响应消息元素

该请求的响应消息中不带消息元素。

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 10.3 复制对象

复制对象(Copy Object)特性用来为OBS上已经存在的对象创建一个副本。

当进行复制对象操作时,目标对象默认复制源对象的元数据;用户也可以将目标对象的元数据替换为本次请求中所带的元数据。新建的目标对象不会复制源对象的ACL信息,默认的新建对象的ACL是private,用户可以使用设置ACL的操作接口来重新设定新对象的ACL。

复制对象操作的请求需要携带鉴权信息,不能携带消息实体。

该操作支持服务端加密功能。

#### 多版本

默认情况下,x-amz-copy-source标识复制源对象的最新版本。如果源对象的最新版本是删除标记,则认为该对象已删除。要复制指定版本的对象,可以在x-amz-copy-source请求消息头中携带versionId参数。

如果目标对象的桶的多版本状态是开启的,系统为目标对象生成唯一的版本号(此版本号与源对象的版本号不同),并且会在响应报头x-amz-version-id返回该版本号。如果目标对象的桶的多版本状态是暂停的,则目标对象的版本号为**null**。



#### 注意

在桶没有开启多版本的情况下,将源对象objecta复制为目标对象objectb,如果在复制操作之前对象objectb已经存在,复制操作执行之后老对象objectb则会被新复制对象objectb覆盖,复制成功后,只能下载到新的对象objectb,老对象objectb将会被删除。因此在使用copy接口时请确保目标对象不存在或者已无价值,避免因copy导致数据误删除。复制过程中源对象objecta无任何变化。

复制对象的结果不能仅根据HTTP返回头域中的status\_code来判断请求是否成功,头域中status\_code返回200时表示服务端已经收到请求,且开始处理复制对象请求。复制是否成功会在响应消息的body中,只有body体中有ETag标签才表示成功,否则表示复制失败。

## 归档存储对象

如果源对象是归档存储对象,需要判断源对象的取回状态,只有当源对象处于已取回状态时,才能复制成功。源对象未取回或者正在取回时,会复制失败,返回错误403 Forbidden。异常描述 为:

ErrorCode: InvalidObjectState

ErrorMessage: Operation is not valid for the source object's storage class

#### 请求消息样式

PUT /destinationObjectName HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com
x-amz-copy-source: /sourceBucket/sourceObject
x-amz-metadata-directive: metadata\_directive

x-amz-copy-source-if-match: etag
x-amz-copy-source-if-none-match: etag

x-amz-copy-source-if-unmodified-since: time\_stamp x-amz-copy-source-if-modified-since: time\_stamp

Authorization: signature

Date: date

## 请求消息参数

该请求消息中不使用消息参数。

# 请求消息头

该消息可以带附加的消息头指定复制的信息,具体如公共请求消息头所示。

#### 表 10-6 请求消息头

消息头名称	描述	是否 必选
x-amz-acl	复制对象时,可以加上此消息头设置对象的权限控制策略,使用的策略为预定义的常用策略,包括: private; public-read; public-read- write; authenticated-read; bucket-owner- read; bucket-owner-full-control。 类型: 字符串。 示例: x-amz-acl: acl	否
x-amz-copy-source	用来指定复制对象操作的源桶名以及源对象名。当源对象存在多个版本时,通过versionId参数指定版本源对象。 类型:字符串。 示例: x-amz-copy-source:/source_bucket/sourceObject	是
x-amz-metadata- directive	此参数用来指定新对象的元数据是从源对象中复制,还是用请求中的元数据替换。 类型:字符串。 有效取值:COPY或REPLACE。 默认值:COPY。 示例:x-amz-metadata-directive:metadata_directive 约束条件:如果此参数的值不是COPY或REPLACE,则OBS立即返回400错误;如果用户进行修改元数据操作(源对象与目标对象相同),则此参数只能为REPLACE,否则此请求作为无效请求,服务端响应400。此参数不支持将加密的对象更改成非加密对象(源对象与目标对象相同)。如果用户使用此参数更改加密的对象,系统将返回400。	否
x-amz-copy-source- if-match	只有当源对象的Etag与此参数指定的值相等时才进行复制对象操作,否则返回412(前置条件不满足)。 类型:字符串。 示例:x-amz-copy-source-if-match: etag 约束条件:此参数可与x-amz-copy-source-if-unmodified-since—起使用,但不能与其它条件复制参数—起使用。	否

消息头名称	描述	是否 必选
x-amz-copy-source- if-none-match	只有当源对象的Etag与此参数指定的值不相等时才进行复制对象操作,否则返回412(前置条件不满足)。 类型:字符串。 示例: x-amz-copy-source-if-none-match: etag	否
	约束条件:此参数可与x-amz-copy-source-if-modified-since一起使用,但不能与其它条件复制参数一起使用。	
x-amz-copy-source- if-unmodified- since	只有当源对象在此参数指定的时间之后没有修改过才进行复制对象操作,否则返回412(前置条件不满足),此参数可与x-amz-copy-source-if-match一起使用,但不能与其它条件复制参数一起使用。 类型:符合http://www.ietf.org/rfc/rfc2616.txt规定格式的HTTP时间字符串。	否
	示例: x-amz-copy-source-if-unmodified -since: time-stamp	
x-amz-copy-source- if-modified-since	只有当源对象在此参数指定的时间之后修改过才进行复制对象操作,否则返回412(前置条件不满足),此参数可与x-amz-copy-source-if-none-match一起使用,但不能与其它条件复制参数一起使用。	否
	类型: 符合http://www.ietf.org/rfc/rfc2616.txt规定格式的 HTTP时间字符串。	
	示例: x-amz-copy-source-if-modified-since: time-stamp	
x-amz-storage-class	复制对象时,可以加上此头域设置目的对象的存储类型。如果未设置此头域,则以目的对象所在桶的默认存储类型作为对象的存储类型。	否
	类型: 字符串	
	说明:存储类型有3种:STANDARD(标准存储)、STANDARD_IA(低频访问存储)、GLACIER(归档存储)。因此这里可配置的值有:STANDARD、STANDARD_IA和GLACIER,注意大小写敏感。	
	示例: x-amz-storage-class: STANDARD	
x-amz-website- redirect-location	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,OBS将这 个值从头域中取出,保存在对象的元数据中。 类型:字符串	否
	<sup>                                    </sup>	
	约束: 必须以 "/" 、 "http://" 或 "https://" 开头, 长度不超过2K。	

消息头名称	描述	是否 必选
x-amz-server-side- encryption	使用该头域表示服务端加密是SSE-KMS方式。目标对象使用SSE-KMS方式加密。 类型:字符串 示例: x-amz-server-side-encryption: aws:kms	否。 当用 SSE- KMS 方时, 必选。
x-amz-server-side- encryption-aws- kms-key-id	SSE-KMS方式下使用该头域,该头域表示加密目标对象使用的主密钥,如果用户没有提供该头域,那么默认的主密钥将会被使用。 类型:字符串示例:x-amz-server-side-encryption-aws-kms-key-id:arn:aws:kms:sichuan:domainiddomainiddomainiddoma0001:key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0	否
x-amz-server-side- encryption- customer-algorithm	SSE-C方式下使用该头域,该头域表示加密目标对象使用的算法。 类型:字符串 示例: x-amz-server-side-encryption-customer-algorithm: AES256 约束:需要和x-amz-server-side-encryption-customer-key, x-amz-server-side-encryption-customer-key-MD5一起使用。	否当用 SSE- C式时必选。
x-amz-server-side- encryption- customer-key	SSE-C方式下使用该头域,该头域表示加密目标对象使用的密钥。 类型:字符串 示例:x-amz-server-side-encryption-customer- key:K7QkYpBkM5+hcs27fsNkUnNVaobncnLht/rCB2o/ 9Cw= 约束:该头域由256-bit的密钥经过base64-encoded得到,需要和x-amz-server-side-encryption-customer- algorithm,x-amz-server-side-encryption-customer- key-MD5一起使用。	否。使 用 SSE- C 式 时 必 选。

消息头名称	描述	是否 必选
x-amz-server-side- encryption- customer-key-MD5	SSE-C方式下使用该头域,该头域表示加密目标对象使用的密钥的MD5值。MD5值用于验证密钥传输过程中没有出错。 类型:字符串示例:x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==约束:该头域由密钥的128-bit MD5值经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-key一起使用。	否当用 SSE- C式时必选。
x-amz-copy-source- server-side- encryption- customer-algorithm	SSE-C方式下使用该头域,该头域表示解密源对象使用的算法。 类型:字符串 示例: x-amz-copy-source-server-side-encryption- customer-algorithm: AES256 约束: 需要和x-amz-copy-source-server-side-encryption- customer-key, x-amz-copy-source-server-side-encryption- customer-key-MD5一起使用。	否当贝对使 SSE方 C式时必选。
x-amz-copy-source- server-side- encryption- customer-key	SSE-C方式下使用该头域,该头域表示解密源对象使用的密钥。用于解密源对象。 类型:字符串 示例: x-amz-copy-source-server-side-encryption- customer-key: K7QkYpBkM5+hcs27fsNkUnNVaobncnLht/rCB2o/9Cw= 约束:该头域由256-bit的密钥经过base64-encoded得 到,需要和x-amz-copy-source-server-side-encryption- customer-algorithm,x-amz-copy-source-server-side- encryption-customer-key-MD5一起使用。	否当贝对使SSC式式时必选。拷源象用E-方方
x-amz-copy-source- server-side- encryption- customer-key-MD5	SSE-C方式下使用该头域,该头域表示解密源对象使用的密钥的MD5值。MD5值用于验证密钥传输过程中没有出错。 类型:字符串示例:x-amz-copy-source-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==约束:该头域由密钥的128-bit MD5值经过base64-encoded得到,需要和x-amz-copy-source-server-side-encryption-customer-algorithm,x-amz-copy-source-server-side-encryption-customer-algorithm,x-amz-copy-source-server-side-encryption-customer-key一起使用。	否当贝对使 SSE方 , 。 拷源象用 E- 方

其他消息头请参见公共请求消息头章节。

#### 请求消息元素

该请求在消息中不使用消息元素。

#### 响应消息样式

HTTP/1.1 status\_code

Content-Type: application/xml

Date: date

Content-Length: length

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<CopyObjectResult xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">

<LastModified>modifiedDate/LastModified>

<ETag>etagValue</ETag>

</CopyObjectResult>

## 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

除公共响应消息头之外,还可能使用如下表10-7中的消息头。

#### 表 10-7 附加响应消息头

消息头名称	描述
x-amz-copy-source-version-id	源对象的版本号。 类型:字符串。
x-amz-version-id	目标对象的版本号。 类型:字符串
x-amz-server-side-encryption	如果服务端加密是SSE-KMS方式,响应包含该头域。 类型:字符串 示例: x-amz-server-side-encryption: aws:kms
x-amz-server-side-encryption-aws- kms-key-id	如果服务端加密是SSE-KMS方式,响应包含该头域,该头域表示主密钥。 示例: x-amz-server-side-encryption-aws-kms-key-id: arn:aws:kms:sichuan:domainiddomainiddoma0001:key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0
x-amz-server-side-encryption- customer-algorithm	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示加密使用的算法。 类型:字符串 示例: x-amz-server-side-encryption-customer- algorithm: AES256

消息头名称	描述
x-amz-server-side-encryption- customer-key-MD5	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示加密使用的密钥的MD5值。 类型:字符串 示例: x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==
x-amz-storage-class	对象为非标准存储对象时,会返回此头域,可取值为: STANDARD_IA或者GLACIER 类型: 字符串

#### 响应消息元素

该请求的响应消息通过消息元素来返回复制结果,元素具体含义如表10-8所示。

#### 表 10-8 响应消息元素

元素名称	描述
CopyObjectResult	复制对象结果的Container。 类型: XML。
LastModified	对象上次修改的时间。 类型:字符串。
ETag	新对象的ETag值。 类型:字符串。

#### 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 10.4 设置对象 ACL

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)。

本节将介绍如何更改对象ACL,改变对象的访问权限。

## 多版本

默认情况下,更改的是最新版本的对象ACL。要设置指定版本的对象ACL,请求可以带参数versionId。

#### 请求消息格式

```
PUT /ObjectName?acl HTTP/1.1
Host: bucketname.obs.cn-north-1.myhwclouds.com
Date: date
Authorization: authorization
<AccessControlPolicy>
   <0wner>
       <ID>ID</ID>
       <DisplayName>displayname
    </0wner>
   <AccessControlList>
       <Grant>
           <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
xsi:type="CanonicalUser">grantee</Grantee>
           <Permission>permission
       </Grant>
   </AccessControlList>
</AccessControlPolicy>
```

## 请求消息参数

请求参数说明如表10-9所示。

#### 表 10-9 请求消息参数

参数名称	描述	是否必选
versionId	对象的版本号。标示更改指定版本的对象ACL。	否
	类型:字符串	

#### 请求消息头

该请求使用公共请求消息头,具体参见公共请求消息头章节。

# 请求消息元素

该请求消息通过带消息元素来传递对象的ACL信息,元素的意义如表10-10所示。

#### 表 10-10 请求消息元素

元素名称	描述	是否必选
ID	用户DomainId。 类型:字符串。	否
DisplayName	用户名称。 类型:字符串。	否
Permission	授予权限。 类型: 枚举类型。	否

#### 响应消息样式

HTTP/1.1 status\_code Content-Length: length Content-Type: application/xml

#### 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

除公共响应消息头之外,还可能使用如表10-11中的消息头。

#### 表 10-11 附加响应消息头

消息头名称	描述
x-amz-version-id	被更改ACL的对象的版本号。
	类型:字符串

#### 响应消息元素

该请求的响应消息中不带有消息元素。

#### 错误响应消息

该请求的响应无特殊错误,所有错误已经包含在表4-1中。

# 10.5 获取对象内容

GET操作从对象存储服务中下载对象。使用GET接口前,请确认必须拥有对象的READ 权限。如果对象Owner授予对匿名用户READ访问权限,则访问该对象

可以不提供鉴权信息。

## 服务端加密

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。

# 多版本

默认情况下,获取的是最新版本的对象。如果最新版本的对象是删除标记,则返回对象不存在。如果要获取指定版本的对象,请求可携带versionId消息参数。

## 归档存储对象

如果要下载的对象是归档存储类对象,由于对象存储在存档设备中,您必须先使用对象取回,然后才能下载该归档存储对象。对象处于不同的取回状态时,给出不同响应:如果对象已取回,下载对象成功时需要返回x-amz-restore头域指示取回失效时间。对未取回或正在取回的归档存储对象发送下载请求时,会返回错误403 Forbidden。

#### 请求消息样式

GET / ObjectName HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization Range:bytes=byte\_range <Optional Additional Header>

#### 🔲 说明

其中Range字段可选,如果没有的话得到全部内容。

#### 请求消息参数

GET操作获取对象内容时,允许用户通过请求参数的方式对一些消息头值进行重写,可以重写的消息头有:Content-Type、Content-Language、Expires、Cache-Control、Content-Disposition以及Content-Encoding共6个。另外所需取回的对象拥有多个版本时,可以通过versionId参数,指定需要下载的版本。具体的说明如表10-12所示。

#### 表 10-12 请求消息参数

参数名称	描述	是否必选
response-content-type	重写响应中的Content-Type头。 类型:字符串。	否
response-content-language	重写响应中的Content-Language头。 类型:字符串。	否
response-expires	重写响应中的Expires头。 类型:字符串。	否
response-cache-control	重写响应中的Cache-Control头。 类型:字符串。	否
response-content-disposition	重写响应中的Content-Disposition 头。 类型:字符串。	否
response-content-encoding	重写响应中的Content-Encoding头。 类型:字符串。	否
versionId	指定获取对象的版本号。 类型:字符串。	否
x-image-process	图片处理服务。 示例: 命令方式: x-image-process=image/ commands 样式方式: x-image-process=style/ stylename 详见《图片处理特性指南》。	沿

## 请求消息头

该请求除使用公共消息头外,还可以使用附加的消息头来完成获取对象的功能,消息 头的意义如**表10-13**所示。

#### 表 10-13 请求消息头

消息头名称	描述	是否必选
Range	获取对象时获取在Range范围内的对象内容;如果Range不合法则忽略此字段下载整个对象;Range是一个范围,它的起始值最小为0,最大为对象长度减1。	否
	类型:字符串。	
	(bytes=byte_range	
	例: bytes=0-4或 bytes=512-1024, 或者表示多个区间bytes=10-20,30-40)。	
If-Modified-Since	如果对象在请求中指定的时间之后有修改,则返回对象内容;否则的话返回304(not modified)。	否
	类型:符合http://www.ietf.org/rfc/ rfc2616.txt规定格式的HTTP时间字符 串。	
If-Unmodified-Since	如果对象在请求中指定的时间之后没有修改,则返回对象内容;否则的话返回412(precondition failed)。 类型:符合http://www.ietf.org/rfc/	否
	rfc2616.txt规定格式的HTTP时间字符 串。	
If-Match	如果对象的ETag和请求中指定的ETag相同,则返回对象内容,否则的话返回412(precondition failed)。	否
	类型:字符串。 (ETag值,例:	
	0f64741bf7cb1089e988e4585d0d3434。)	
If-None-Match	如果对象的ETag和请求中指定的ETag不相同,则返回对象内容,否则的话返回304(not modified)。	否
	类型:字符串。	
	(ETag值,例: 0f64741bf7cb1089e988e4585d0d3434。)	

消息头名称	描述	是否必选
x-amz-server-side- encryption-customer- algorithm	SSE-C方式下使用该头域,该头域表示加密使用的算法。 类型:字符串	否。当使用SSE-C 方式时,必选。
	示例: x-amz-server-side-encryption- customer-algorithm: AES256	
	约束: 需要和x-amz-server-side- encryption-customer-key, x-amz-server- side-encryption-customer-key-MD5一起使 用。	
x-amz-server-side- encryption-customer- key	SSE-C方式下使用该头域,该头域表示加密使用的密钥。该密钥用于解密对象。 类型:字符串 示例: x-amz-server-side-encryption-customer- key:K7QkYpBkM5+hcs27fsNkUnNVaobnc	否。当使用SSE-C 方式时,必选。
	nLht/rCB2o/9Cw= 约束: 该头域由256-bit的密钥经过 base64-encoded得到,需要和x-amz- server-side-encryption-customer- algorithm,x-amz-server-side-encryption- customer-key-MD5一起使用。	
x-amz-server-side- encryption-customer- key-MD5	SSE-C方式下使用该头域,该头域表示加密使用的密钥的MD5值。MD5值用于验证密钥传输过程中没有出错。 类型:字符串示例:x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==约束:该头域由密钥的128-bit MD5值经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-	否。当使用SSE-C 方式时,必选。
	algorithm, x-amz-server-side-encryption-customer-key一起使用。	

# 请求消息元素

该请求消息中不使用消息元素。

# 响应消息样式

HTTP/1.1 status\_code Content-Type: type Date: date

Content-Length: length

Etag: etag

Last-Modified: time <Object Content>

## 响应消息头

该请求的响应消息使用公消息头,具体请参考公共响应消息头章节。

除公共响应消息头之外,还可能使用如下表10-14中的消息头。

#### 表 10-14 附加响应消息头

消息头名称	描述	
x-amz-expiration	如果对象配置了过期时间,那么响应的消息中应该包含此消息头。该消息头用expiry-date和rule-id两个键值对描述了对象的详细过期信息。 类型:字符串。	
x-amz-website-redirect-location	当桶设置了Website配置,就可以设置对象元数据的这个属性,Website接入点返回301重定向响应,将请求重定向到该属性指定的桶内的另一个对象或外部的URL。 类型:字符串	
x-amz-delete-marker	标识对象是否是删除标记。如果不是,则响应中不会出现该消息头。 类型:布尔值 有效值: true false 默认值: false	
x-amz-version-id	对象的版本号。如果该对象无版本号,则响应中不会出现该消息头。 有效值:字符串 默认值:无	
x-amz-server-side-encryption	如果服务端加密是SSE-KMS方式,响应包含该头域。 类型:字符串 示例: x-amz-server-side-encryption: aws:kms	
x-amz-server-side-encryption-aws- kms-key-id	如果服务端加密是SSE-KMS方式,响应包含该头域,该头域表示主密钥。 示例: x-amz-server-side-encryption-aws-kms-key-id: arn:aws:kms:sichuan:domainiddomainiddomainiddoma0001:key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0	

消息头名称	描述
x-amz-server-side-encryption- customer-algorithm	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示解密使用的算法。
	类型:字符串
	示例: x-amz-server-side-encryption-customer- algorithm: AES256
x-amz-server-side-encryption- customer-key-MD5	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示解密使用的密钥的MD5值。
	类型:字符串
	示例: x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==

#### 响应消息元素

该请求的响应消息中不带消息元素。

#### 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 10.6 获取对象元数据

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。

#### 多版本

默认情况下,获取的是最新版本的对象元数据。如果最新版本的对象是删除标记,则返回404。如果要获取指定版本的对象元数据,请求可携带versionId消息参数。

## 请求消息样式

HEAD /ObjectName HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

请求参数说明如表10-15所示。

#### **表 10-15** 请求消息参数

参数名称	描述	是否必选
versionId	对象的版本号。	否
	类型:字符串	

## 请求消息头

该请求使用公共消息头,具体请参考**公共请求消息头**章节。 另外该请求可以使用附加的消息头,具体如**表10-16**所示。

#### 表 10-16 请求消息头

消息头名称	描述	是否必选
Origin	预请求指定的跨域请求Origin(通常为域名)。 类型:字符串	是
Access- Control- Request- Headers	实际请求可以带的HTTP头域,可以带多个头域。 类型:字符串	否
x-amz-server- side- encryption- customer- algorithm	SSE-C方式下使用该头域,该头域表示解密使用的算法。 类型:字符串 示例: x-amz-server-side-encryption-customer- algorithm: AES256 约束: 需要和x-amz-server-side-encryption-customer- key, x-amz-server-side-encryption-customer- key. MD5—起使用。	否。当使用 SSE-C方式 时,必选。
x-amz-server- side- encryption- customer-key	SSE-C方式下使用该头域,该头域表示解密使用的密钥。 类型:字符串示例:x-amz-server-side-encryption-customer-key:K7QkYpBkM5+hcs27fsNkUnNVaobncnLht/rCB2o/9Cw=约束:该头域由256-bit的密钥经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-key-MD5一起使用。	否。当使用 SSE-C方式 时,必选。

消息头名称	描述	是否必选
x-amz-server- side- encryption- customer-key- MD5	SSE-C方式下使用该头域,该头域表示解密使用的密钥的MD5值。MD5值用于验证密钥传输过程中没有出错。 类型:字符串示例:x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==约束:该头域由密钥的128-bit MD5值经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-key一起使用。	否。当使用 SSE-C方式 时,必选。

# 请求消息元素

该请求消息中不使用消息元素。

# 响应消息样式

HTTP/1.1 status\_code
Content-Type: type
Date: date

Content-Length: *length* Etag: *etag* 

Last-Modified: time

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

除公共响应消息头之外,还可能使用如下表10-17中的消息头。

#### 表 10-17 附加响应消息头

消息头名称	描述
x-amz-expiration	如果对象配置了过期时间,那么响应的消息中应该包含此消息头。该消息头用expiry-date和rule-id两个键值对描述了对象的详细过期信息。 类型:字符串。
x-amz-website-redirect-location	当桶设置了Website配置,就可以设置对象元数据的这个属性,Website接入点返回301重定向响应,将请求重定向到该属性指定的桶内的另一个对象或外部的URL。 类型:字符串

消息头名称	描述
x-amz-version-id	对象的版本号。如果该对象无版本号,则响应中不会出现该消息头。 类型:字符串 默认值:无
Access-Control-Allow-Origin	当桶设置了CORS配置,如果请求的Origin满足服务端的CORS配置,则在响应中包含这个Origin。 类型:字符串
Access-Control-Allow-Headers	当桶设置了CORS配置,如果请求的headers满足服务端的CORS配置,则在响应中包含这个headers。 类型:字符串
Access-Control-Max-Age	当桶设置了CORS配置,服务端CORS配置中的 MaxAgeSeconds。 类型:整数
Access-Control-Allow-Methods	当桶设置了CORS配置,如果请求的Access-Control-Request-Method满足服务端的CORS配置,则在响应中包含这条rule中的Methods。 类型:字符串有效值:GET、PUT、HEAD、POST、DELETE
Access-Control-Expose-Headers	当桶设置了CORS配置,服务端CORS配置中的 ExposeHeader。 类型:字符串
x-amz-server-side-encryption	如果服务端加密是SSE-KMS方式,响应包含该头域。 类型:字符串 示例: x-amz-server-side-encryption: aws:kms
x-amz-server-side-encryption-aws- kms-key-id	如果服务端加密是SSE-KMS方式,响应包含该头域,该头域表示主密钥。 示例: x-amz-server-side-encryption-aws-kms-key-id: arn:aws:kms:sichuan:domainiddomainiddomainiddoma0001:key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0
x-amz-server-side-encryption- customer-algorithm	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示解密使用的算法。 类型:字符串 示例: x-amz-server-side-encryption-customer- algorithm: AES256

消息头名称	描述
x-amz-server-side-encryption- customer-key-MD5	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示解密使用的密钥的MD5值。 类型:字符串 示例: x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==
x-amz-storage-class	对象为非标准存储对象时,会返回此头域,可取值为: STANDARD_IA或者GLACIER 类型: 字符串
x-amz-restore	对象为归档存储类型,并且处于正在取回或已经取回时,会返回此头域。表示对象的取回状态:正在取回 ongoing-request="true";已取回 ongoing-request="false", expiry-date="Wed, 07 Nov 2012 00:00:00 GMT"。其中expiry-date表示对象取回后的失效时间。

# 响应消息元素

该请求的响应消息中不带消息元素。

# 错误响应消息

无特殊错误; 所有错误已经包含在表4-1中。

# 10.7 获取对象 ACL

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。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。

# 多版本

默认情况下,获取最新版本的对象ACL。如果最新版本的对象是删除标记,则返回404。如果要获取指定版本的对象ACL,请求可携带versionId消息参数。

# 请求消息样式

GET /ObjectName?acl HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

该请求需要在请求消息参数中指定是在获取对象ACL,参数意义如表10-18所示。

#### 表 10-18 请求消息参数

参数名称	描述	是否必选
acl	指定该请求是获取对象的acl。 类型:字符串。	是
versionId	指定对象的版本号。	否

# 请求消息头

该请求使用公共消息头,具体请参考公共请求消息头章节。

# 请求消息元素

该请求消息中不使用消息元素。

## 响应消息样式

```
HTTP/1.1 status_code
Date: date
Content-Length: length
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AccessControlPolicy xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">
    <0wner>
        <ID>id</ID>
        \verb|\langle DisplayName \rangle| name \\ |\langle DisplayName \rangle|
    </0wner>
    \langle AccessControlList \rangle
        <Grant>
            xsi:type="CanonicalUser">
                 <ID>id</ID>
                <DisplayName>name/DisplayName>
             </Grantee>
            <Permission>permission
        {\  \, \langle /{\tt Grant} \rangle \  \, }
    </AccessControlList>
<\!/\texttt{AccessControlPolicy}\!>
```

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

除公共响应消息头之外,还可能使用如下表10-19中的消息头。

#### 表 10-19 附加响应消息头

消息头名称	描述
x-amz-version-id	指定对象的版本号。
	有效值:字符串
	默认值: 无

# 响应消息元素

该请求的响应消息中通过消息元素返回对象的ACL信息,元素的具体意义如**表10-20**所示。

#### 表 10-20 响应消息元素

元素名称	描述
DisplayName	用户的名称。 类型:字符串。
ID	用户的DomainId。 类型:字符串。
AccessControlList	访问控制列表,记录了对该桶有访问权限的用户列表。 类型: XML。
Grant	用于标记用户及用户的权限。 类型:XML。
Grantee	记录用户信息。 类型: XML。
Permission	指定的用户对该对象所具有的操作权限。 类型:字符串。

# 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 10.8 删除对象

删除对象的操作。如果要删除的对象不存在,则仍然返回成功信息。

# 多版本

当桶的多版本状态是开启时,不指定版本删除对象将产生一个带唯一版本号的删除标记,并不删除对象;当桶的多版本状态是Suspended时,不指定版本删除将1删除版本号为null的对象,并将产生一个版本号为null的删除标记。

如果要删除指定版本的对象,请求可携带versionId消息参数。

# 请求消息样式

DELETE /ObjectName HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

请求参数说明如表10-21所示。

#### **表 10-21** 请求消息参数

参数名称	描述	是否必选
versionId	待删除对象的版本号。	否
	类型:字符串	

# 请求消息头

该请求使用公共消息头,具体请参考公共请求消息头章节。

# 请求消息元素

该请求消息中不使用消息元素。

# 响应消息样式

HTTP/1.1 status\_code
Date: date

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

除公共响应消息头之外,还可能使用如下表10-22中的消息头。

#### 表 10-22 附加响应消息头

消息头名称	描述
x-amz-delete-marker	标识对象是否标记删除。如果不是,则响应中不 会出现该消息头。
	类型: 布尔值
	有效值: true false
	默认值: false
x-amz-version-id	对象的版本号。如果该对象无版本号,则响应中 不会出现该消息头。
	有效值:字符串
	默认值:无

## 响应消息元素

该请求的响应消息中不带消息元素。

## 错误响应消息

无特殊错误,所有错误已经包含在表4-1中。

# 10.9 批量删除对象

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。

批量删除对象支持两种响应方式: verbose和quiet。Verbose是指在返回响应时,不管对象是否删除成功都将删除结果包含在XML响应里; quiet是指在返回响应时,只返回删除失败的对象结果,没有返回的认为删除成功。OBS默认使用verbose模式,如果用户在请求消息体中指定quiet模式的话,使用quiet模式。

批量删除的请求消息头中必须包含Content-MD5以及Content-Length,用以保证请求的消息体在服务端检测到网络传输如果有错,则可以检测出来。

## 请求消息样式

```
POST /?delete HTTP/1.1
Host: bucketname. obs. cn-north-1. myhwclouds. com
Authorization: authorization
Content-MD5: MD5
Content-Length: length
<?xml version="1.0" encoding="UTF-8"?>
<Delete>
    <Quiet> true</Quiet>
    <0bject>
         <Kev>Kev</Kev>
         <VersionId>VersionId/VersionId>
    </object>
    <0bject>
         \langle \text{Key} \rangle \textit{Key} \langle /\text{Key} \rangle
    </0biect>
</Delete>
```

# 请求消息参数

该请求的请求消息中不使用消息参数。

# 请求消息头

该请求使用公共消息头,具体请参考公共请求消息头章节。

## 请求消息元素

该请求通过在请求消息的消息元素中指定要批量删除的对象列表,元素的具体含义如表10-23所示。

#### 表 10-23 请求消息元素

元素名称	描述	是否必须
Quiet	用于指定使用quiet模式,只返回删除失败的对象结果;如果有此字段,则必被置为True,如果为False则被系统忽略掉。 类型:布尔值。	否
Delete	待删除的对象列表。 类型: XML。	是
Object	待删除的对象。 类型: XML。	是
Key	待删除的对象Key。 类型:字符串。	是
VersionId	待删除的对象版本号。 类型:字符串。	否



## 注意

请求约束

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)。

# 响应消息样式

```
HTTP/1.1 status_code
Date: date
Content-Type: application/xml
Content-Length: length

Content-Length: length

Content-Length: length

Content-Length: length

ColleteResult xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">

Cheleted
```

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应通过消息元素返回删除的结果,元素的具体意义如表10-24中所示。

#### 表 10-24 响应消息元素

字段名称	描述	是否必选
DeleteResult	批删响应的根节点。 类型:容器。	是
Deleted	删除成功结果的Container。 类型:容器。	是
Error	删除失败结果的Container。 类型:容器。	是
Key	每个删除结果的对象名。 类型:字符串。	是
Code	删除失败结果的错误码。 类型:字符串。	是
Message	删除失败结果的错误消息。 类型:字符串。	是
VersionId	删除对象的版本号 类型:字符串。	否
DeleteMarker	当批量删除请求访问的桶是多版本桶时,如果创建或删除一个删除标记,返回消息中该元素的值为true。 类型:布尔值。	否
DeleteMarkerVersi onId	请求创建或删除的删除标记版本号。 当批量删除请求访问的桶是多版本桶时,如果创建或删除一个删除标记,响应消息会返回该元素。该元素在以下两种情况中会出现:  • 用户发送不带版本删除请求,即请求只有对象名,无版本号。这种情况下,系统会创建一个删除标记,并在响应中返回该删除标记的版本号。  • 用户发送带版本删除请求,即请求同时包含对象名以及版本号,但是该版本号标识一个删除标记。这种情况下,系统会删除此删除标记。这种情况下,系统会删除此删除标记,并在响应中返回该删除标记的版本号。 类型: 布尔值。	否

## 错误响应消息

- 1、用户收到请求后首先进行XML的解析,如果超过1000个对象返回400 Bad Request。
- 2、如果XML消息体中包含的对象Key不合法,比如超过1024字节,OBS返回400 Bad Request。
- 3、如果请求消息头中不包含Content-MD5,OBS则返回400 Bad Request。

其他错误已经包含在表4-1中。

# 10.10 取回归档存储对象

如果要获取归档存储对象的内容,需要先将对象取回。然后再执行下载数据的操作。

## 多版本

默认情况下,取回的是最新版本的对象。如果最新版本的对象是删除标记,则返回404。如果要取回指定版本的对象,请求可携带versionId消息参数。

## 请求消息样式

POST /ObjectName?restore&versionId=VersionID HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization string

Content-MD5: MD5

<RestoreRequest>

<Days>NumberOfDays<GlacierJobParameters>

<Tier>RetrievalOption</Tier>

</GlacierJobParameters>

</RestoreRequest>

# 请求消息参数

参数名称	描述	是否必选
versionId	待取回归档存储对象的版本号。	否
	类型:字符串	

# 请求消息头

该请求使用公共消息头,具体请参见公共请求消息头章节。

# 请求消息元素

#### 表 10-25 请求消息元素表

名字	描述	是否必选
RestoreRequest	取回信息的容器 类型:容器	是
Days	取回对象的保存时间,最小值为1,最大值为30。 类型:正整数	是
GlacierJobParameters	取回选项的容器 类型:容器	否
Tier	取回选项,支持两种取值: [Expedited Standard]。 Expedited表示快速取回对象,取回耗时1~5 min, Standard表示标准取回对象,取回耗时3~5 h, 默认取值为Standard。 类型: 字符串	否

# 响应消息样式

HTTP/1.1 status\_code
Date: date

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应消息中不带有响应元素。

# 错误响应消息

#### 表 10-26 对象存储访问服务错误码列表

错误码	描述	HTTP状态码
RestoreAlreadyInPro	对象正在取回,请求冲突	409 Conflict
gress	ErrorMessage: Object restore is already in	
	progress	

错误码	描述	HTTP状态码
ObjectHasAlreadyRe stored	已经取回的对象,禁止缩短取回保存时 间。	409 Conflict
	ErrorMessage:After restoring an archived object, you cannot shorten the restoration period of the archived object	
MalformedXML	Days字段不合法(不为整数)	400 Bad Request
	ErrorMessage: The XML you provided was not well-formed or did not validate against our published schema	
InvalidArgument	Days字段取值超出范围(1<=days<=30)	400 Bad Request
	ErrorMessage: restoration days should be at least 1 and at most 30	
MalformedXML	Tier字段不合法	400 Bad Request
	ErrorMessage: The XML you provided was not well-formed or did not validate against our published schema	
InvalidObjectState	取回对象不是归档存储对象	403 Forbidden
	ErrorMessage: Restore is not allowed, as object's storage class is not GLACIER	

# 11 多段上传

- 11.1 初始化上传段任务
- 11.2 列举多段上传任务
- 11.3 上传段
- 11.4 拷贝段
- 11.5 列举已上传的段
- 11.6 合并段
- 11.7 取消多段上传任务

# 11.1 初始化上传段任务

使用多段上传特性时,用户必须首先调用创建多段上传任务接口创建任务,系统会给用户返回一个全局唯一的多段上传任务号,作为任务标识。后续用户可以根据这个标识发起相关的请求,如:上传段、合并段、列举段等。创建多段上传任务不影响已有的同名对象;同一个对象可以同时存在多个多段上传任务;每个多段上传任务在初始化时可以附加消息头信息,包括acl、用户自定义元数据和通用的HTTP消息头contentType、contentEncoding等,这些附加的消息头信息将先记录在多段上传任务元数据中。

该操作支持服务端加密功能。

## 请求消息样式

POST /ObjectName?uploads HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

该请求需要在消息中指定参数,表明这是多段上传,参数意义如表11-1所示。

## 表 11-1 请求消息参数

参数名称	描述	是否必选
uploads	表明这是多段上传任务。	是
	类型:字符串。	

# 请求消息头

该请求可以使用附加的消息头,具体如表11-2所示。

## 表 11-2 请求消息头

消息头名称	描述	是否 必选
x-amz-storage-class	初始化多段上传任务时,可以加上此头域设置对象的存储类型。如果未设置此头域,则以桶的默认存储类型作为对象的存储类型。 类型:字符串 说明:存储类型有3种:STANDARD(标准存储)、 STANDARD_IA(低频访问存储)、GLACIER(归档存储)。因此这里可配置的值有:STANDARD、 STANDARD_IA和GLACIER,注意大小写敏感。 示例:x-amz-storage-class:STANDARD	否
x-amz-website- redirect-location	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,OBS将这个值从头域中取出,保存在对象的元数据中。 类型:字符串 默认值:无 约束:必须以"/"、"http://"或"https://"开头,长度不超过2K。	否
x-amz-server-side- encryption	使用该头域表示服务端加密是SSE-KMS方式。 类型:字符串 示例:x-amz-server-side-encryption:aws:kms	否。 当用 SSE- KMS 方时 必选。

消息头名称	描述	是否 必选
x-amz-server-side- encryption-aws- kms-key-id	SSE-KMS方式下使用该头域,该头域表示主密钥,如果用户没有提供该头域,那么默认的主密钥将会被使用。 类型:字符串 示例: x-amz-server-side-encryption-aws-kms-key-id: arn:aws:kms:sichuan:domainiddomainiddoma0001 :key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0	否
x-amz-server-side- encryption- customer-algorithm	SSE-C方式下使用该头域,该头域表示加密使用的算法。 类型:字符串 示例: x-amz-server-side-encryption-customer-algorithm: AES256 约束: 需要和x-amz-server-side-encryption-customer- key, x-amz-server-side-encryption-customer- key. D5— 起使用。	否当用 SSE- C式时必选。
x-amz-server-side- encryption- customer-key	SSE-C方式下使用该头域,该头域表示加密使用的密钥。该密钥用于加密对象。 类型:字符串示例:x-amz-server-side-encryption-customer-key:K7QkYpBkM5+hcs27fsNkUnNVaobncnLht/rCB2o/9Cw=约束:该头域由256-bit的密钥经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-key-MD5一起使用。	否当用 SSE C 式时必选。
x-amz-server-side- encryption- customer-key-MD5	SSE-C方式下使用该头域,该头域表示加密使用的密钥的MD5值。MD5值用于验证密钥传输过程中没有出错。类型:字符串示例:x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==约束:该头域由密钥的128-bit MD5值经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-key一起使用。	否当用 SSE- C式时必选。
x-amz-acl	初始化多段上传任务时,可以加上此消息头设置对象的权限控制策略,使用的策略为预定义的常用策略,包括: private; public-read; public-read-write; authenticated-read; bucket-owner-read; bucket-owner-full-control。 类型:字符串。 示例: x-amz-acl: private	否

消息头名称	描述	是否 必选
x-amz-meta-*	初始化多段上传任务时,可以在HTTP请求中加入以 "x-amz-meta-"开头的消息头,用来加入自定义的元数 据,以便对对象进行自定义管理。当用户获取此对象或 查询此对象元数据时,加入的自定义元数据将会在返回 消息的头中出现。HTTP请求不包含消息体,长度不能 超过8KB。 类型:字符串 示例: x-amz-meta-test: test metadata	否

其他公共消息头请参考公共请求消息头章节。

## 请求消息元素

该请求消息中不使用消息元素。

# 响应消息样式

HTTP/1.1 status\_code

Date: date

Content-Length: *length* Connection: *status* 

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<InitiateMultipartUploadResult xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">

<Bucket>BucketName

<Key>ObjectName</Key>

<UploadId>uploadID</UploadId>

</InitiateMultipartUploadResult>

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 表 11-3 附加响应头消息

消息头名称	描述
x-amz-server-side-encryption	如果服务端加密是SSE-KMS方式,响应包含该头域。 类型:字符串示例: x-amz-server-side-encryption:aws:kms

消息头名称	描述
x-amz-server-side-encryption-aws-kms-key-id	如果服务端加密是SSE-KMS方式,响应 包含该头域,该头域表示主密钥。
	示例: x-amz-server-side-encryption-aws-kms-key-id: arn:aws:kms:sichuan:domainiddomainiddomainiddomainiddomainiddoma0001:key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0
x-amz-server-side-encryption-customer- algorithm	如果服务端加密是SSE-C方式,响应包含 该头域,该头域表示加密使用的算法。 类型:字符串
	示例: x-amz-server-side-encryption- customer-algorithm: AES256
x-amz-server-side-encryption-customer- key-MD5	如果服务端加密是SSE-C方式,响应包含 该头域,该头域表示加密使用的密钥的 MD5值。
	类型:字符串
	示例: x-amz-server-side-encryption- customer-key-MD5:4XvB3tbNTN+tIEVa0/ fGaQ==

# 响应消息元素

该请求响应消息中通过返回消息元素,返回本次多段上传任务的多段上传任务号、桶名、对象名,供后续上传段、合并段使用,元素的具体意义如**表11-4**所示。

表 11-4 响应消息元素

元素名称	描述
InitiateMultipartUpload Result	描述多段上传任务的容器。 类型:XML。
Bucket	多段上传对象所在桶的桶名。 类型:字符串。
Key	多段上传对象的key。 类型:字符串。
UploadId	多段上传id,后面进行多段上传时,利用这个id指定多段上 传任务。 类型:字符串。

# 错误响应消息

- 1、如果AccessKey或签名无效,OBS返回403 Forbidden,错误码为AccessDenied。
- 2、如果桶不存在,OBS返回404 Not Found,错误码为NoSuchBucket。
- 3、检查用户是否具有指定桶的写权限,如果没有权限,OBS返回403 Forbidden,错误码为AccessDenied。

其他错误已包含在表4-1中。

# 11.2 列举多段上传任务

用户可以通过"列出多段上传任务"接口查询一个桶中所有的初始化后还未合并以及 未取消的多段上传任务。

# 请求消息样式

GET /?uploads&max-uploads=max HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Authorization: authorization

# 请求消息参数

该请求可以通过在请求消息中指定参数,查询指定范围的多段上传任务,请求参数说明如**表11-5**所示。

#### 表 11-5 请求消息参数

参数名称	描述	是否必选
delimiter	对于名字中包含delimiter的对象的任务,其对象名(如果请求中指定了prefix,则此处的对象名需要去掉prefix)中从首字符至第一个delimiter之间的字符串将作为CommonPrefix在响应中返回。对象名包含CommonPrefix的任务被视为一个分组,作为一条记录在响应中返回,该记录不包含任务的信息,仅用于提示用户该分组下存在多段上传任务。类型:字符串。	否
prefix	如果请求中指定了prefix,则响应中仅包含对象名以prefix开始的任务信息。 类型:字符串。	否
max-uploads	列举的多段任务的最大条目,取值范围为[1,1000], 当超出范围时,按照默认的1000进行处理。 类型:整型。	否
key-marker	列举时返回指定的key-marker之后的多段任务。 类型:字符串。	否

参数名称	描述	是否必选
upload-id-marker	只有和key-marker一起使用才有意义, 列举时返回 指定的key-marker的upload-id-marker之后的多段任 务。 类型:字符串。	否

# 请求消息头

该请求使用公共请求消息头,具体请见公共请求消息头章节。

# 请求消息元素

该请求消息中不使用消息元素。

# 响应消息样式

```
HTTP/1.1 status_code
Date: date
Content-Length: length
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ListMultipartUploadsResult xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">
               <Bucket>bucketname
               <KeyMarker/>
               <UploadIdMarker/>
               <NextKeyMarker>nextMarker/NextKeyMarker>
               <NextUploadIdMarker>idMarker/NextUploadIdMarker>
               <MaxUploads>maxUploads
               \langle IsTruncated \rangle true \langle /IsTruncated \rangle
               <Upload>
                              <Key>key</Key>
                              <UploadId>uploadID/UploadId>
                              <Initiator>
                                              <ID>id</ID>
                                              <DisplayName>name
                              </Initiator>
                              <0wner>
                                              <ID>ownerID</ID>
                                              <DisplayName>OwnerDisplayName
                              \verb| StorageClass| | storageCl
                               <Initiated>initiatedDate/Initiated>
              </Upload>
</ListMultipartUploadsResult>
```

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应消息中通过消息元素返回多段上传任务,元素的具体意义如表11-6所示。

## 表 11-6 响应消息元素

元素名称	描述	是否必选
ListMultipartUploads Result	保存List Multipart Upload请求结果的容器。 类型:容器。 子节点: Bucket, KeyMarker, UploadIdMarker, NextKeyMarker, NextUploadIdMarker, MaxUploads, Delimiter, Prefix, Upload, CommonPrefixes, IsTruncated。 父节点: None。	是
Bucket	初始化任务所在的桶名。 类型:字符串。 父节点:ListMultipartUploadsResult。	是
KeyMarker	列举时的起始对象位置。 类型:字符串。 父节点: ListMultipartUploadsResult。	是
UploadIdMarker	列举时的起始UploadId位置。 类型:字符串。 父节点:ListMultipartUploadsResult。	是
NextKeyMarker	如果本次没有返回全部结果,响应请求中将包含NextKeyMarker字段,用于标明接下来请求的KeyMarker值。 类型:字符串。 父节点: ListMultipartUploadsResult。	是
NextUploadIdMarker	如果本次没有返回全部结果,响应请求中将包含NextUploadMarker字段,用于标明接下来请求的UploadMarker值。 类型: 字符串。 父节点: ListMultipartUploadsResult。	是
MaxUploads	返回的最大多段上传任务数目。 类型:整型。 父节点: ListMultipartUploadsResult。	是
IsTruncated	表明是否本次返回的Multipart Upload结果列表被截断。"true"表示本次没有返回全部结果; "false"表示本次已经返回了全部结果. 类型: Boolean。 父节点: ListMultipartUploadsResult。	是

元素名称	描述	是否必选
Upload	保存Multipart Upload任务信息的容器。 类型:容器。 子节点: Key, UploadId, InitiatorOwner, StorageClass,Initiated。 父节点: ListMultipartUploadsResult。	是
Key	初始化Multipart Upload任务的Object名字。 类型:字符串。 父节点: Upload。	是
UploadId	Multipart Upload任务的ID。 类型: 字符串。 父节点: Upload。	是
Initiator	Multipart Upload任务的创建者。 子节点: ID, DisplayName。 类型:容器。 父节点: Upload。	是
ID	创建者的DomainId。 类型:字符串。 父节点: Initiator, Owner。	是
DisplayName	创建者的名字。 类型:字符串。 父节点: Initiator, Owner。	是
Owner	段的所有者。 类型:容器。 子节点: ID, DisplayName。 父节点: Upload。	是
StorageClass	表明待多段上传的对象存储类型。 类型: 枚举值。 父节点: Upload。	是
Initiated	Multipart Upload任务的初始化时间。 类型: Date。 父节点: Upload。	是
ListMultipartUploads Result.Prefix	请求中带的Prefix。 类型:字符串。 父节点: ListMultipartUploadsResult。	是

元素名称	描述	是否必选
Delimiter	请求中带的Delimiter。 类型:字符串。 父节点: ListMultipartUploadsResult。	是
CommonPrefixes	请求中带Delimiter参数时,返回消息带CommonPrefixes分组信息。 类型:容器。 父节点: ListMultipartUploadsResult。	是
CommonPrefixes. Prefix	CommonPrefixes分组信息中,表明不同的Prefix。 类型:字符串。 父节点:CommonPrefixes。	是

## 错误响应消息

1、OBS系统对maxUploads进行判断,如果maxUploads不为整数类型或者小于0,OBS返回400 Bad Request。

其他错误已经包含在表4-1中。

# 11.3 上传段

多段上传任务创建后,用户可以通过指定多段上传任务号,通过上传段接口为特定的任务上传段,从客户端上传新数据。同一个对象的同一个多段上传任务在上传段时,上传的顺序对后续的合并操作没有影响,也即支持多个段并发上传。

段大小范围是[5MB, 5GB],但在进行合并段操作时,最后一个段的大小范围为[0,5GB]。上传的段的编号也有范围限制,其范围是[1,10000]。

该操作支持服务端加密功能。



#### 注意

多并发上传同一对象的同一段时,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名段上传的并发性。同一对象的不同段并发上传不需要加锁。

# 请求消息样式

PUT /ObjectName?partNumber=partNum&uploadId=uploadID HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

Content-Length: length
Authorization: authorization

Content-MD5: md5
<object Content>

# 请求消息参数

在上传段的时候需要通过在消息参数中指定多段上传任务号和段号来上传指定段,参数的具体意义如**表11-7**所示。

表 11-7 请求消息参数

参数名称	描述	是否必选
partNumber	上传段的段号。取值为从1到10000的整数 类型:整型。	是
uploadId	多段上传任务Id。 类型:字符串。	是

# 请求消息头

该请求使用公共消息头,具体请参考公共请求消息头章节。

表 11-8 服务端加密请求消息头

消息头名称	描述	是否必选
x-amz-server-side- encryption-customer- algorithm	SSE-C方式下使用该头域,该头域表示加密使用的算法。 类型:字符串示例:x-amz-server-side-encryption-customer-algorithm:AES256约束:需要和x-amz-server-side-encryption-customer-key,x-amz-server-side-encryption-customer-key-MD5一起使	否。当使用SSE-C方式时,必选,需要与初始化上传段任务使用相同的算法。
	用。	

消息头名称	描述	是否必选
x-amz-server-side- encryption-customer-key	SSE-C方式下使用该头域,该头域表示加密使用的密钥。该密钥用于加密对象。 类型:字符串示例:x-amz-server-side-encryption-customer-key:K7QkYpBkM5+hcs27fsNkUnNVaobncnLht/rCB2o/9Cw=约束:该头域由256-bit的	否。当使用SSE-C方式时,必选,需要与初始化上传段任务使用相同的密钥。
	密钥经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-key-MD5一起使用。	
x-amz-server-side- encryption-customer-key- MD5	SSE-C方式下使用该头域,该头域表示加密使用的密钥的MD5值。MD5值用于验证密钥传输过程中没有出错。 类型:字符串示例:x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==	否。当使用SSE-C方式时,必选,需要与初始化上传段任务使用相同的MD5值。
	约束: 该头域由密钥的 128-bit MD5值经过base64- encoded得到,需要和x- amz-server-side-encryption- customer-algorithm,x- amz-server-side-encryption- customer-key一起使用。	

# 请求消息元素

该请求消息中不使用消息元素。

# 响应消息样式

HTTP/1.1 status\_code

Date: date ETag: etag

Content-Length: length

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 表 11-9 附加响应头消息

消息头名称	描述
x-amz-server-side-encryption	如果服务端加密是SSE-KMS方式,响应包含该头域。 类型:字符串示例: x-amz-server-side-encryption:aws:kms
x-amz-server-side-encryption-aws-kms-key-id	如果服务端加密是SSE-KMS方式,响应包含该头域,该头域表示主密钥。示例: x-amz-server-side-encryption-aws-kms-key-id: arn:aws:kms:sichuan:domainiddomai
x-amz-server-side-encryption-customer- algorithm	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示加密使用的算法。 类型:字符串 示例:x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer- key-MD5	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示加密使用的密钥的MD5值。 类型:字符串示例:x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==

# 响应消息元素

该请求的响应消息中不带消息元素。

# 错误响应消息说明

- 1、如果段序号超过范围[1,10000],则返回错误400 Bad Request。
- 2、如果段大小超过5G,则返回错误400 Bad Request。
- 3、如果AccessKey或签名无效,OBS返回403 Forbidden,错误码为AccessDenied。
- 4、查询桶是否存在,如果桶不存在,OBS返回404 Not Found,错误码为NoSuchBucket。

- 5、检查桶的ACL,判断用户DomainId是否具有指定桶的写权限,如果没有权限,则OBS返回403 Forbidden,错误码为AccessDenied。
- 6、检查多段上传任务是否存在,如果不存在,OBS返回404 Not Found,错误码为NoSuchUpload。
- 7、检查请求用户是否是多段上传任务的发起者(Initiator),如果不是,OBS返回403 Forbidden,错误码为AccessDenied。

其他错误已包含在表4-1中。

# 11.4 拷贝段

多段上传任务创建后,用户可以通过指定多段上传任务号,为特定的任务上传段。添加段的方式还包括调用段拷贝接口。允许客户将已上传对象的一部分或全部拷贝为段。

该操作支持服务端加密功能。



#### 注意

拷贝段的结果不能仅根据HTTP返回头域中的status\_code来判断请求是否成功,头域中status\_code返回200时表示服务端已经收到请求,且开始处理拷贝段请求。拷贝是否成功会在响应消息的body中,只有body体中有ETag标签才表示成功,否则表示拷贝失败。

将源对象object拷贝为一个段part1,如果在拷贝操作之前part1已经存在,拷贝操作执行之后老段数据part1会被新拷贝的段数据覆盖。拷贝成功后,只能列举到最新的段part1,老段数据将会被删除。因此在使用拷贝段接口时请确保目标段不存在或者已无价值,避免因拷贝段导致数据误删除。拷贝过程中源对象object无任何变化。

## 归档存储对象

如果源对象是归档存储对象,需要判断源对象的取回状态,只有当源对象处于已取回状态时,才能拷贝成功。源对象未取回或者正在取回时,会拷贝失败,返回错误403 Forbidden。异常描述为:

ErrorCode: InvalidObjectState

ErrorMessage: Operation is not valid for the source object's storage class

## 请求消息样式

PUT /ObjectName?partNumber=partNum&uploadId=UploadID HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date

x-amz-copy-source: sourceobject
x-amz-copy-source-range:bytes=start-end

Authorization: authorization Content-Length: length

# 请求消息参数

拷贝段需要在参数中指定目标段的段号和多段上传任务号,参数的具体意义如**表11-10** 所示。

#### **表 11-10** 请求消息参数

参数名称	描述	是否必选
partNumber	上传段的段号。 类型:整型。	是
uploadId	多段上传任务Id。 类型:字符串。	是

# 请求消息头

该请求的除了使用公共消息头外,还使用了两个扩展的消息头,如**公共请求消息头**所示。

## 表 11-11 请求消息头

消息头名称	描述	是否必选
x-amz-copy-source	拷贝的源对象。 类型:字符串。	是
x-amz-copy-source-range	源对象中待拷贝的段的字节范围(start - end), start为段起始字节,end为 段结束字节。 类型:整型。	否
x-amz-server-side- encryption-customer- algorithm	SSE-C方式下使用该头域,该头域表示加密目标段使用的算法。 类型:字符串示例:x-amz-server-side- encryption-customer- algorithm: AES256 约束:需要和x-amz- server-side-encryption- customer-key, x-amz- server-side-encryption- customer-key-MD5—起使用。	否。当使用SSE-C方式时,必选,需要与初始化上传段任务使用相同的算法。

消息头名称	描述	是否必选
x-amz-server-side- encryption-customer-key	SSE-C方式下使用该头域,该头域表示加密目标段使用的密钥。该密钥用于加密对象。 类型:字符串示例:x-amz-server-side-encryption-customer-key:K7QkYpBkM5+hcs27fsNkUnNVaobncnLht/rCB2o/9Cw=约束:该头域由256-bit的密钥经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-side-encryption-customer-	否。当使用SSE-C方式时,必选,需要与初始化上传段任务使用相同的密钥。
x-amz-server-side- encryption-customer-key- MD5	key-MD5一起使用。  SSE-C方式下使用该头域,该头域表示加密目标段使用的密钥的MD5值。 MD5值用于验证密钥传输过程中没有出错。 类型:字符串示例:x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==约束:该头域由密钥的128-bit MD5值经过base64-encoded得到,需要和x-amz-server-side-encryption-customer-algorithm,x-amz-server-side-encryption-customer-key—起使用。	否。当使用SSE-C方式时,必选,需要与初始化上传段任务使用相同的MD5值。

消息头名称	描述	是否必选
x-amz-copy-source-server-side-encryption-customer-algorithm	SSE-C方式下使用该头域,该头域表示源对象使用的算法。 类型:字符串示例:x-amz-copy-source-server-side-encryption-customer-algorithm: AES256 约束:需要和x-amz-copy-source-server-side-encryption-customer-key,x-amz-copy-source-server-side-encryption-customer-key-MD5一起使用。	否。当拷贝源对象使用 SSE-C方式时,必选。
x-amz-copy-source-server-side-encryption-customer-key	SSE-C方式下使用该头域,该头域表示源对象使用的密钥。用于解密源对象。 类型:字符串示例:x-amz-copy-source-server-side-encryption-customer-key: K7QkYpBkM5+hcs27fsNk UnNVaobncnLht/rCB2o/9Cw=约束:该头域由256-bit的密钥经过base64-encoded得到,需要和x-amz-copy-source-server-side-encryption-customer-algorithm,x-amz-copy-source-server-side-encryption-customer-algorithm,x-amz-copy-source-server-side-encryption-customer-key-MD5一起使用。	否。当拷贝源对象使用 SSE-C方式时,必选。

消息头名称	描述	是否必选
x-amz-copy-source-server- side-encryption-customer- key-MD5	SSE-C方式下使用该头域,该头域表示源对象使用的密钥的MD5值。MD5值用于验证密钥传输过程中没有出错。 类型:字符串示例:x-amz-copy-source-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==约束:该头域由密钥的	否。当拷贝源对象使用 SSE-C方式时,必选。
	128-bit MD5值经过base64-encoded得到,需要和x-amz-copy-source-server-side-encryption-customer-algorithm,x-amz-copy-source-server-side-encryption-customer-key一起使用。	

# 请求消息元素

该请求消息中不使用消息元素。

# 响应消息样式

HTTP/1.1 status\_code

Date: date

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CopyPartResult xmlns="http://obs.cn-north-1.myhwclouds.com/doc/2015-06-30/">

<LastModified>modifiedDate/LastModified>

<ETag>*etag*</ETag> </CopyPartResult>

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

#### 表 11-12 附加响应头消息

消息头名称            描	描述
包   类   示	如果服务端加密是SSE-KMS方式,响应 包含该头域。 类型:字符串 示例: x-amz-server-side-encryption:

消息头名称	描述
x-amz-server-side-encryption-aws-kms-key-id	如果服务端加密是SSE-KMS方式,响应 包含该头域,该头域表示主密钥。
	示例: x-amz-server-side-encryption-aws- kms-key-id: arn:aws:kms:sichuan:domainiddomainiddo mainiddoma0001:key/4f1cd4de- ab64-4807-920a-47fc42e7f0d0
x-amz-server-side-encryption-customer- algorithm	如果服务端加密是SSE-C方式,响应包含 该头域,该头域表示加密使用的算法。 类型:字符串
	示例: x-amz-server-side-encryption- customer-algorithm: AES256
x-amz-server-side-encryption-customer- key-MD5	如果服务端加密是SSE-C方式,响应包含 该头域,该头域表示加密使用的密钥的 MD5值。
	类型:字符串
	示例: x-amz-server-side-encryption- customer-key-MD5:4XvB3tbNTN+tIEVa0/ fGaQ==

# 响应消息元素

该请求的响应消息使用消息元素来返回段拷贝的结果,元素的意义如表11-13所示。

#### 表 11-13 响应消息元素

元素名称	描述
LastModified	对象上次修改时间。 类型:字符串。
ETag	目标段的ETag值。 类型:字符串。

# 错误响应消息

- 1、如果AccessKey或签名无效,OBS返回403 Forbidden,错误码为AccessDenied。
- 2、查询源桶或目的桶是否存在,如果不存在,OBS返回404 Not Found,错误码为NoSuchBucket。
- 3、如果源对象不存在,OBS返回404 Not Found,错误码为NoSuchKey。
- 4、如果用户对指定对象没有读权限,OBS返回403 Forbidden,错误码为AccessDenied。

- 5、如果用户对目的桶没有写权限,OBS返回403 Forbidden,错误码为AccessDenied。
- 6、查询指定的任务不存在,OBS返回404 Not Found,错误码为NoSuchUpload。
- 7、如果用户不是多段上传任务的发起者,OBS返回403 Forbidden,错误码为 AccessDenied。
- 8、当拷贝的单段超过5G时,OBS返回400 Bad Request。
- 9、如果段序号超过范围[1,10000], OBS返回错误400 Bad Request。

其他错误已包含在表4-1中。

# 11.5 列举已上传的段

用户可以通过"列出段"接口查询一个任务所属的所有段信息。

## 请求消息样式

GET /ObjectName?uploadId=uploadid&max-parts=max&part-number-marker=marker HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: date Authorization: auth

## 请求消息参数

该请求通过请求消息参数指定多段上传任务以及列出的段数量,参数的具体含义如**表11-14**所示。

#### 表 11-14 请求消息参数

参数名称	描述	是否必选
uploadId	多段上传任务的id。 类型:字符串。 默认值:无。	是
max-parts	规定在列举已上传段响应中的最大Part数目。 类型:字符串。 默认值:1,000。	否
part-number -marker	指定List的起始位置,只有Part Number数目大于该参数的Part会被列出。 类型:字符串。 默认值:无。	否

## 请求消息头

该请求使用公共消息头,具体请参考公共请求消息头章节。

## 请求消息元素

该请求消息中不使用消息元素。

# 响应消息样式

```
HTTP/1.1 status_code
Date: date
Content-Length: length
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Key>object</Key>
   <UploadId>uploadid/UploadId>
    <Initiator>
       <ID>id</ID>
       <DisplayName>displayname
    </Initiator>
    <0wner>
       <ID>ownerid</ID>
       \verb|\langle DisplayName \rangle| ownername \\ <| DisplayName \rangle|
    \verb| StorageClass| storageclass| | StorageClass| |
    <PartNumberMarker>partNmebermarker/PartNumberMarker>
    <NextPartNumberMarker>nextPartnumberMarker/NextPartNumberMarker>
    <MaxParts>maxParts
    <IsTruncated>true</IsTruncated>
    (Part)
        <PartNumber>partNumber
       <LastModified>modifiedDate/LastModified>
       \langle \text{ETag} \rangle \textit{etag} \langle / \text{ETag} \rangle
       ⟨Size⟩size⟨/Size⟩
   </Part>
</ListPartsResult>
```

# 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

# 响应消息元素

该请求的响应通过消息元素返回已上传了的段信息,元素的具体含义如表11-15所示。

#### 表 11-15 响应消息元素

响应字段名称	描述
ListPartsResult	保存List Part请求结果的容器。 类型:容器。 子节点: Bucket, Key, UploadId, PartNumberMarker, NextPartNumberMarker, MaxParts, IsTruncated, Part。 父节点:无。
Bucket	Bucket名称。 类型:字符串。 父节点: ListPartsResult。
Key	Object名称。 类型:字符串。 父节点: ListPartsResult。

响应字段名称	描述
UploadId	Upload任务ID。 类型:字符串。 父节点:ListPartsResult。
Initiator	Upload任务的创建者。 类型:容器。 子节点: ID、DisplayName。 父节点: ListPartsResult。
Owner	和Initiator相同(和S3兼容,S3在使用IAM的情况下,Initiator和Owner可能不同)。 类型:容器。 子节点: ID, DisplayName。 父节点: ListPartsResult。
ID	创建者的DomainId。 类型:字符串。 父节点: Initiator、Owner。
DisplayName	创建者的名字。 类型:字符串。 父节点: Initiator、Owner。
StorageClass	存储类型。 类型: 枚举值 有效值: STANDARD   STANDARD_IA   GLACIER 父节点: ListPartsResult。
PartNumberMarker	本次List结果的Part Number起始位置。 类型:整数。 父节点: ListPartsResult。
NextPartNumberMar ker	如果本次没有返回全部结果,响应请求中将包含 NextPartNumberMarker元素,用于标明接下来请求的 PartNumberMarker值。 类型:整数。 父节点: ListPartsResult。
MaxParts	返回请求中最大的Part数目。 类型:整数。 父节点: ListPartsResult。

响应字段名称	描述
IsTruncated	标明是否本次返回的List Part结果列表被截断。"true"表示本次没有返回全部结果;"false"表示本次已经返回了全部结果。 类型: boolean。
	父节点: ListPartsResult。
Part	保存Part信息的容器。 类型:字符串。 子节点:PartNumber, LastModified, ETag, Size。 父节点:ListPartsResult。 (PartNumber 标示Part的数字。)
PartNumber	已上传Part的编号。 类型: 整型。 父节点: ListPartsResult.Part。
LastModified	Part上传的时间。 类型: 日期。 父节点: ListPartsResult.part。
ETag	已上传Part内容的ETag。 类型:字符串。 父节点:ListPartsResult.Part
Size	已上传Part大小。 类型:整数。 父节点: ListPartsResult.Part。

## 错误响应消息

- 1、如果AccessKey或签名无效,OBS返回403 Forbidden,错误码为AccessDenied。
- 2、如果请求的桶不存在,OBS返回404 Forbidden,错误码为NoSuchBucket。
- 3、如果请求的多段上传任务不存在,OBS返回404 Not Found,错误码为NoSuchUpload。
- 4、OBS判断用户DomainId是否具有指定桶的读权限,如果没有权限,则OBS返回403 Forbidden,错误码为AccessDenied。

其他错误已经包含在表4-1中。

# 11.6 合并段

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"完成后删除多余的段数据,且同时释放容量配额。

合并完成的多段上传数据可以通过已有的下载对象接口,下载整个多段上传对象或者指定Range下载整个多段上传对象的某部分数据。

合并完成的多段上传数据可以通过已有的删除对象接口,删除整个多段上传对象的所 有分段数据,删除后不可恢复。

合并完成的多段上传数据不记录整个对象的MD5作为Etag,在下载多段数据或List桶内对象看到的多段数据其Etag的生成方式为: MD5 ( $M_1M_2$  ······ $M_N$ ) -N, 其中, $M_n$ 表示第n段的MD5值,如11.6示例所示,有3个分段,每个分段都有对应的MD5值,合并段ETag的生成是先将3个分段的MD5合并起来再进行MD5计算得到一个新值,再拼接-N作为合并段的ETag值,-N表示总共有多少段,在该示例中即为-3。

## 多版本

如果桶的多版本状态是开启的,则合并段后得到的对象生成一个唯一的版本号,并且会在响应报头x-amz-version-id返回该版本号。如果桶的多版本状态是暂停的,则合并段后得到的对象版本号为null。关于桶的多版本状态,参见8.6 设置桶的多版本状态。



## 注意

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。

# 请求消息样式

```
POST /ObjectName?uploadId=uploadID HTTP/1.1
Host: bucketname.obs.cn-north-1.myhwclouds.com
Date: date
Content-Length: length
Authorization: authorization
<CompleteMultipartUpload>
   (Part)
       <PartNumber>partNum
       <ETag>etag</ETag>
   </Part>
       <PartNumber>partNum
       <ETag>etag</ETag>
   </Part>
   <Part>
       <PartNumber>partNum
       <ETag>etag</ETag>
</CompleteMultipartUpload>
```

## 请求消息参数

该请求在消息参数中指定多段上传任务号来标明它要合并哪一个上传段任务,参数意义如**表11-16**所示。

## 表 11-16 请求消息参数

参数名称	描述	是否必选
uploadId	指明多段上传任务。	是
	类型:字符串。	

## 请求消息头

该请求使用公共消息头,具体请参考公共请求消息头章节。

## 请求消息元素

该请求需要在消息中带消息元素,指定要合并的段列表,元素的具体意义如**表11-17**中 所示

#### 表 11-17 请求消息元素

元素名称	描述	是否必选
CompleteMultipart Upload	合并的段列表。 类型:XML。	是
PartNumber	段号。 类型:整型。	是
ETag	对应段的ETag值。 类型:字符串。	是

## 响应消息样式

HTTP/1.1 status\_code

Date: date

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Key>ObjectName</Key>

<ETag>ETag</ETag>

</CompleteMultipartUploadResult>

## 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

除公共响应消息头之外,还可能使用如表11-18中的消息头。

## 表 11-18 附加响应消息头

消息头名称	描述
x-amz-version-id	合并得到的对象的版本号。 类型:字符串
x-amz-server-side-encryption	如果服务端加密是SSE-KMS方式,响应包含该头域。 类型:字符串 示例: x-amz-server-side-encryption: aws:kms
x-amz-server-side-encryption-aws- kms-key-id	如果服务端加密是SSE-KMS方式,响应包含该头域,该头域表示主密钥。 示例: x-amz-server-side-encryption-aws-kms-key-id: arn:aws:kms:sichuan:domainiddomainiddomainiddoma0001:key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0
x-amz-server-side-encryption- customer-algorithm	如果服务端加密是SSE-C方式,响应包含该头域,该头域表示加密使用的算法。 类型:字符串 示例: x-amz-server-side-encryption-customer- algorithm: AES256

## 响应消息元素

该请求的响应消息中通过返回消息元素来返回合并段的结果,元素的具体意义如**表11-19**所示。

## 表 11-19 响应消息元素

元素	描述
Location	合并后得到的对象的url。 类型:字符串。
Bucket	合并段所在的桶。 类型:字符串。
Key	合并得到对象的key。 类型:字符串。
ETag	根据各个段的ETag计算得出的结果。 类型:字符串。

## 错误响应消息

- 1、如果没有消息体,OBS返回400 Bad Request。
- 2、如果消息体格式不正确,OBS返回400 Bad Request。
- 3、消息体中如果段信息未按照段序号升序排列,OBS返回400 Bad Request,错误码为InvalidPartOrder。
- 4、如果AccessKey或签名无效,OBS返回403 Forbidden,错误码为AccessDenied。
- 5、如果请求的桶不存在,OBS返回404 Not Found,错误码为NoSuchBucket。
- 6、如果请求的多段上传任务不存在,OBS返回404 Not Found,包含错误信息NoSuchUpload。
- 7、如果用户不是该任务的发起者(initiator),OBS返回403 Forbidden,错误码为 AccessDenied。
- 8、在合并段时如果请求段列表中包含了不存在的段,OBS返回400 Bad Request,错误码为InvalidPart。
- 9、如果请求段列表中包含的段的Etag错误,OBS返回400 Bad Request,错误码为InvalidPart。
- 10、除最后一个段之外的其它段尺寸过小(小于5MB),OBS返回400 Bad Request。
- 11、对象在合并完成后总大小如果超过48.8TB,OBS返回400 Bad Request。

其他错误已包含在表4-1中。

## 11.7 取消多段上传任务

如果用户希望取消一个任务,可以调用取消多段上传任务接口取消任务。合并段或取消任务接口被调用后,用户不能再对任务进行上传段和列举段的操作。

## 请求消息样式

DELETE /ObjectName?uploadId=uplaodID HTTP/1.1

Host: bucketname.obs.cn-north-1.myhwclouds.com

Date: *date*Authorization: *auth* 

## 请求消息参数

该请求通过消息参数,指定要取消的段任务的多段上传任务号,参数的意义如**表11-20** 所示。

### 表 11-20 请求消息参数

参数名称	描述	是否必选
uploadId	指明多段上传任务。	是
	类型:字符串。	

## 请求消息头

该请求使用公共消息头,具体请参考公共请求消息头章节。

## 请求消息元素

该请求消息中不使用消息元素。

## 响应消息样式

 $\begin{array}{ll} {\tt HTTP/1.1~status\_code} \\ {\tt Date:~date} \end{array}$ 

## 响应消息头

该请求的响应消息使用公共消息头,具体请参考公共响应消息头章节。

## 响应消息元素

该请求的响应消息中不带消息元素。

## 错误响应消息

- 1、如果AccessKey或签名无效,OBS返回403 Forbidden,错误码为AccessDenied。
- 2、如果请求的桶不存在,OBS返回404 Not Found,错误码为NoSuchBucket。
- 3、用户执行取消多段上传任务操作时判断用户是否是任务初始化者或是桶的所有者,如果不是则OBS则返回403 Forbidden。
- 4、操作成功,OBS向用户返回204 No Content。

其他错误已包含在表4-1中。

## **12** 服务端加密

用户可以使用普通方式上传、下载对象,也可以使用服务端加密方式进行上传、下载对象。

OBS支持服务端加密功能,使加密的行为在服务端进行。

用户可以根据自身的需求,使用不同的密钥管理方式来使用服务端加密功能。当前支持两种服务端加密方式: KMS托管密钥的服务端加密(SSE-KMS)和客户提供加密密钥的服务端加密(SSE-C)。

SSE-KMS方式,OBS使用KMS(Key Management Service)服务提供的密钥进行服务端加密。

SSE-C方式,OBS使用用户提供的密钥和密钥的MD5值进行服务端加密。

使用服务端加密,返回的ETag值不是对象的MD5值。使用服务端加密上传对象,服务端不会校验传入的Content-MD5。

- 12.1 服务端加密SSE-KMS方式
- 12.2 服务端加密SSE-C方式
- 12.3 与服务端加密相关的接口

## 12.1 服务端加密 SSE-KMS 方式

SSE-KMS方式,OBS使用KMS(Key Management Service)服务提供的密钥进行服务端加密。用户首次向区域中的桶上传 SSE-KMS 加密的对象时,OBS将自动为您创建一个默认客户主密钥,客户主密钥用于加密和解密KMS提供的密钥。SSE-KMS方式不支持用户自建密钥。Bucket ACL/Policy不支持SSE-KMS加密对象进行跨租户授权访问。

SSE-KMS方式新增加两个头域来支持SSE-KMS加密。

x-amz-server-side- encryption	使用该头域表示服务端加密是SSE-KMS方式。对象使用SSE-KMS方式加密。	
	示例: x-amz-server-side-encryption: aws:kms	

x-amz-server-sideencryption-aws-kmskey-id SSE-KMS方式下使用该头域,该头域表示加密对象使用的主密钥,如果用户没有提供该头域,那么默认的主密钥将会被使用。

示例: x-amz-server-side-encryption-aws-kms-key-id: arn:aws:kms:regionname:domainid:key/4f1cd4de-ab64-4807-920a-47fc42e7f0d0

## 表 12-1 该新增的两个头域可以应用于如下接口

#### 接口

PUT 上传对象

POST 上传对象

复制对象 (新增的头域针对目标对象)

初始化上传段任务

OBS支持Bucket Policy,如果您要对所有存储在桶中的对象执行服务端加密限制,则可以使用这些策略。例如,如果本租户的上传对象请求不包含用于请求服务端加密 (SSE-KMS) 的头域x-amz-server-side-encryption:"aws:kms",则下面的Bucket Policy将拒绝该上传对象请求。

## 12.2 服务端加密 SSE-C 方式

SSE-C方式,OBS使用用户提供的密钥和密钥的MD5值进行服务端加密。

OBS 不存储您提供的加密密钥,如果您丢失加密密钥,则会无法获取该对象。SSE-C 方式新增加六个头域来支持SSE-C加密。

使用SSE-C方式加密对象, 您必须使用下面的三个头域。

x-amz-server-side-	SSE-C方式下使用该头域,该头域表示加密对象使用的算法。
encryption-customer-	示例: x-amz-server-side-encryption-customer-algorithm:
algorithm	AES256
x-amz-server-side-	SSE-C方式下使用该头域,该头域表示加密对象使用的密钥。
encryption-customer-	示例: x-amz-server-side-encryption-customer-
key	key:K7QkYpBkM5+hcs27fsNkUnNVaobncnLht/rCB2o/9Cw=
x-amz-server-side- encryption-customer- key-MD5	SSE-C方式下使用该头域,该头域表示加密对象使用的密钥的MD5值。MD5值用于验证密钥传输过程中没有出错。示例: x-amz-server-side-encryption-customer-key-MD5:4XvB3tbNTN+tIEVa0/fGaQ==

## 表 12-2 该新增的三个头域可以应用于如下接口

接口
PUT 上传对象
POST 上传对象
复制对象 (新增的头域针对目标对象)
获取对象元数据
获取对象内容
初始化上传段任务
上传段
拷贝段(新增的头域针对目标段)

针对复制对象和拷贝段,另外增加三个头域支持源对象是SSE-C加密的场景。

x-amz-copy-source- server-side- encryption-customer- algorithm	SSE-C方式下使用该头域,该头域表示解密源对象使用的算法。 示例: x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-copy-source- server-side- encryption-customer- key	SSE-C方式下使用该头域,该头域表示解密源对象使用的密钥。用于解密源对象。 示例: x-amz-copy-source-server-side-encryption-customer-algorithm: K7QkYpBkM5+hcs27fsNkUnNVaobncnLht/rCB2o/9Cw=
x-amz-copy-source- server-side- encryption-customer- key-MD5	SSE-C方式下使用该头域,该头域表示解密源对象使用的密钥的MD5值。MD5值用于验证密钥传输过程中没有出错。示例: x-amz-copy-source-server-side-encryption-customer-key: 4XvB3tbNTN+tIEVa0/fGaQ==

## 12.3 与服务端加密相关的接口

本节介绍,与服务端加密相关的接口,以及使用该接口时服务端加密对传输协议和鉴 权的使用要求。

与服务端加密相关的接口对使用传输协议和鉴权有具体要求,详情如表所示。

## 表 12-3 与 SSE-C 相关的接口对传输协议

接口	传输协议
PUT 上传对象	HTTPS
POST 上传对象	HTTPS
初始化上传段任务	HTTPS
获取对象元数据	HTTPS
获取对象内容	HTTPS
上传段	HTTPS
合并段	HTTP or HTTPS

## 表 12-4 与 SSE-KMS 相关的接口对传输协议

接口	传输协议
PUT 上传对象	HTTPS
POST 上传对象	HTTPS
初始化上传段任务	HTTPS
获取对象元数据	HTTP or HTTPS
获取对象内容	HTTPS
上传段	HTTPS
合并段	HTTP or HTTPS

## 表 12-5 复制对象接口对传输协议

源对象	目标对象	传输协议
非加密对象	SSE-KMS加密对象	HTTPS
SSE-KMS加密对象	SSE-KMS加密对象	HTTPS

源对象	目标对象	传输协议
SSE-C加密对象	SSE-KMS加密对象	HTTPS
非加密对象	SSE-C加密对象	HTTPS
SSE-KMS加密对象	SSE-C加密对象	HTTPS
SSE-C加密对象	SSE-C加密对象	HTTPS
非加密对象	非加密对象	HTTP or HTTPS
SSE-KMS加密对象	非加密对象	HTTP or HTTPS
SSE-C加密对象	非加密对象	HTTP or HTTPS

## 表 12-6 拷贝段接口对传输协议

源对象	目标段	传输协议
非加密对象	SSE-KMS加密段	HTTP or HTTPS
SSE-KMS加密对象	SSE-KMS加密段	HTTP or HTTPS
SSE-C加密对象	SSE-KMS加密段	HTTP or HTTPS
非加密对象	SSE-C加密段	HTTPS
SSE-KMS加密对象	SSE-C加密段	HTTPS
SSE-C加密对象	SSE-C加密段	HTTPS
非加密对象	非加密段	HTTP or HTTPS
SSE-KMS加密对象	非加密段	HTTP or HTTPS
SSE-C加密对象	非加密段	HTTP or HTTPS

# 13 并发一致性说明

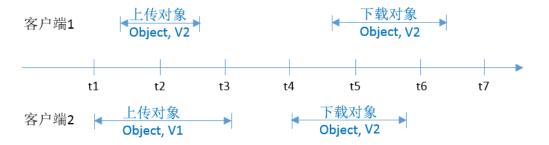
当客户端发起的写/删除请求返回成功之后,客户端可以获取到最新数据。当写操作客户端等待超时、服务端返回500或者503的HTTP响应错误码时,之后的读取操作有可能成功读取到数据,也有可能读不到数据。建议客户端在出现上述错误时,查询数据是否已经上传成功,如果不成功则重新上传。

针对同一个对象或桶的操作,比如多个客户端对同一个对象并行上传、查询和删除时,具体操作结果依赖于操作到达系统的时间和系统内部处理的时延,可能返回不一致的结果。比如,当多个客户端并行上传同一个对象时,系统最后收到的上传请求会覆盖前一个上传的对象。如果需要避免同一个对象被并行访问,需要在上层应用中增加对象的锁机制。

## 并发操作举例

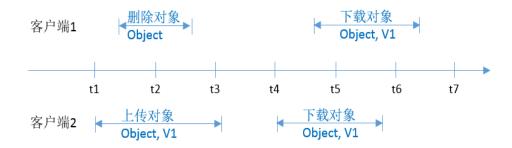
1. 当客户端2正在上传一个对象v1时,客户端1同时上传一个同名的对象v2成功后,不管是客户端1还是客户端2都能够读取最新的对象数据v2,如图13-1所示。

#### 图 13-1 并发成功上传同一个对象



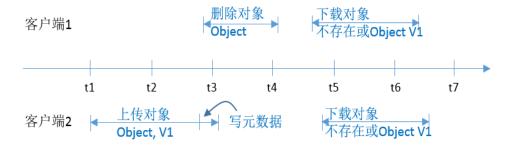
2. 当客户端2上传一个对象v1的时候,如果在对象数据上传且还没有写入对象元数据的过程中,客户端1删除同名的对象成功后,客户端2的上传对象操作仍然返回成功,并且不论客户端1还是客户端2都能读取到对象数据v1,如图13-2所示。

## 图 13-2 并发上传和删除同一个对象(1)



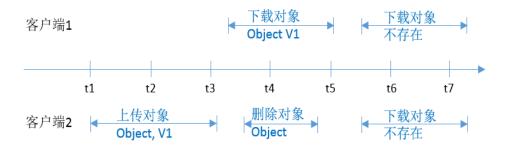
3. 当客户端2上传一个对象v1的时候,如果在对象数据上传完成,系统写入对象元数据的短暂过程中,客户端1发起删除同名的对象成功后,客户端2的上传对象操作仍然返回成功,但是客户端1和客户端2下载对象Object1时,有可能读到对象数据v1,也有可能返回对象不存在,如图13-3所示。

## 图 13-3 并发上传和删除同一个对象(2)



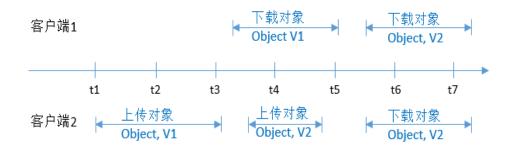
4. 当客户端1下载一个对象的过程中,客户端2发起删除同名对象操作,此时客户端1可能下载到完整的对象数据,也有可能只能下载到对象的一部分数据。当客户端2的删除操作返回成功后,再发起下载对象请求,则返回对象不存在,如图13-4所示。

#### 图 13-4 并发下载和删除同一个对象



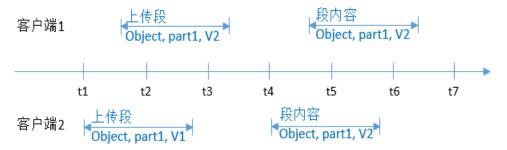
5. 当客户端1下载一个对象的过程中,客户端2发起更新同名对象操作,此时客户端1可能下载到完整的对象数据,也有可能只能下载到对象的一部分数据。当客户端2的更新操作返回成功后,再发起下载对象请求,则返回最新的对象数据,如图13-5所示。

#### 图 13-5 并发下载和更新同一个对象



6. 当客户端2正在上传一个对象的段v1时,客户端1同时上传同一个对象的相同段号的段v2成功后,不管是客户端1还是客户端2列举段时都能够列举etag为v2的段信息,如图13-6所示。

#### 图 13-6 并发上传同名对象的同名段



- 7. 当客户端上传一个对象,客户端等待超时、服务端返回500或503的错误码,在之后的对象下载和列举桶中对象会有如下四种可能:
- (1) 对象上传失败,对象下载和列举桶中对象均无该对象信息。
- (2)客户端等待超时,有极低的概率出现head该对象正常,但是下载该对象报500错误,若遇到此问题,请重新上传该对象。
- (3)客户端上传的是新创建对象,查看桶中对象列表将查询不到该对象,但是通过对象名直接下载,可能下载此对象。
- (4) 客户端上传的是原有对象或同名对象,查看桶中对象列表可以查询到该对象,但查看到的对象的大小、etag、创建时间、owner信息与下载对象时得到的信息不一致。
- 8. 客户端1上传一个对象v1,客户端2同时上传一个同名的对象v2,服务端返回200,在 之后的对象下载和列举桶中对象会有如下四种可能:
- (1) 对象下载和列举桶中对象信息均为对象v1的信息。
- (2) 对象下载和列举桶中对象信息均为对象v2的信息。
- (3)下载对象为v1,列举桶中对象为对象v2的信息。
- (4) 下载对象为v2, 列举桶中对象为对象v1的信息。
- 9. 客户端使用同一用户并发创删同名桶时,有较小概率会发生创删桶冲突,导致请求失败返回500。

## $\mathbf{A}$ $_{$ 文档修订记录

发布日期	修订记录
2018-04-25	第十五次正式发布。 增加跨区域复制特性的描述。
2018-04-17	第十四次正式发布。 增加联邦认证特性的描述。
2018-03-06	第十三次正式发布。 增加关于低频访问存储和归档存储的描述。
2017-12-30	第十二次正式发布。 修正部分易用性问题。
2017-11-10	第十一次正式发布。 修正部分易用性问题。
2017-10-17	第十次正式发布。 修正部分易用性问题。
2017-09-27	第九次正式发布。 优化请求示例中的访问方式,修正一些细节问题。
2017-08-18	第八次正式发布。 优化文档目录结构。

发布日期	修订记录
2017-08-09	第七次正式发布。 增加关于低频访问存储和归档存储的描述。
2017-06-23	第六次正式发布。 新增关于桶策略的示例。
2016-12-09	第五次正式发布。 新增支持事件通知功能。
2016-08-01	第四次正式发布。 新增服务端加密内容。
2016-05-20	第三次正式发布。 修正一些问题。
2015-10-30	第二次正式发布。 修正一些问题。
2015-06-30	第一次正式发布。